Unbiased Learning to Rank: Theory and Practice

Qingyao Ai[1], Jiaxin Mao[2], Yiqun Liu[2], W. Bruce Croft[1]
[1]CICS, University of Massachusetts Amherst
[2]DCST, Tsinghua University

Before the Tutorial

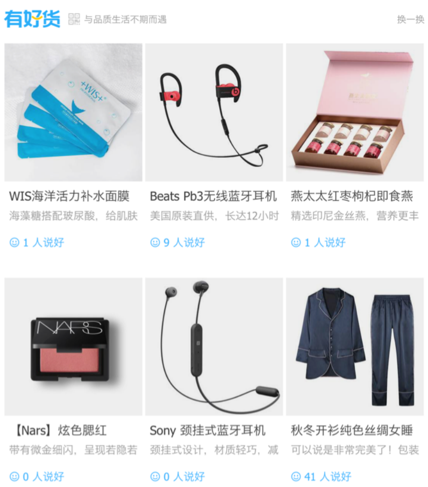
- Tutorial Website
  - Tutorial Resources on the CIKM 2018 Tutorial Page:
    - http://www.cikm2018.units.it/tutorial8.html
  - Google Site:
    - https://sites.google.com/site/tutorialofultrforwebsearch/home
- Slides and Supplemental Materials
  - https://sites.google.com/site/tutorialofultrforwebsearch/Resources
- Q&A Form
  - https://sites.google.com/site/tutorialofultrforwebsearch/qa

Outline

- Introduction
  - Problem Analysis
  - Existing Solutions
- Part 1: Click Models
  - Basic Concept and Hypothesis
  - Advanced Click Model
  - Applications
- Part 2: Unbiased Learning Algorithms
  - Inverse Propensity Weighting
  - Online Result Randomization
  - Simulation/ Real-world Experiments
  - Advanced Topics
- Summary

Introduction

Ranking



Learning to Rank

- Given a learning-to-rank system *S*:

Documents

Features

[0.1, 0.3, …, -0.2]
[-0.3, 0.5, …, -0.8]
[0.4, 0.2, …, -0.1]

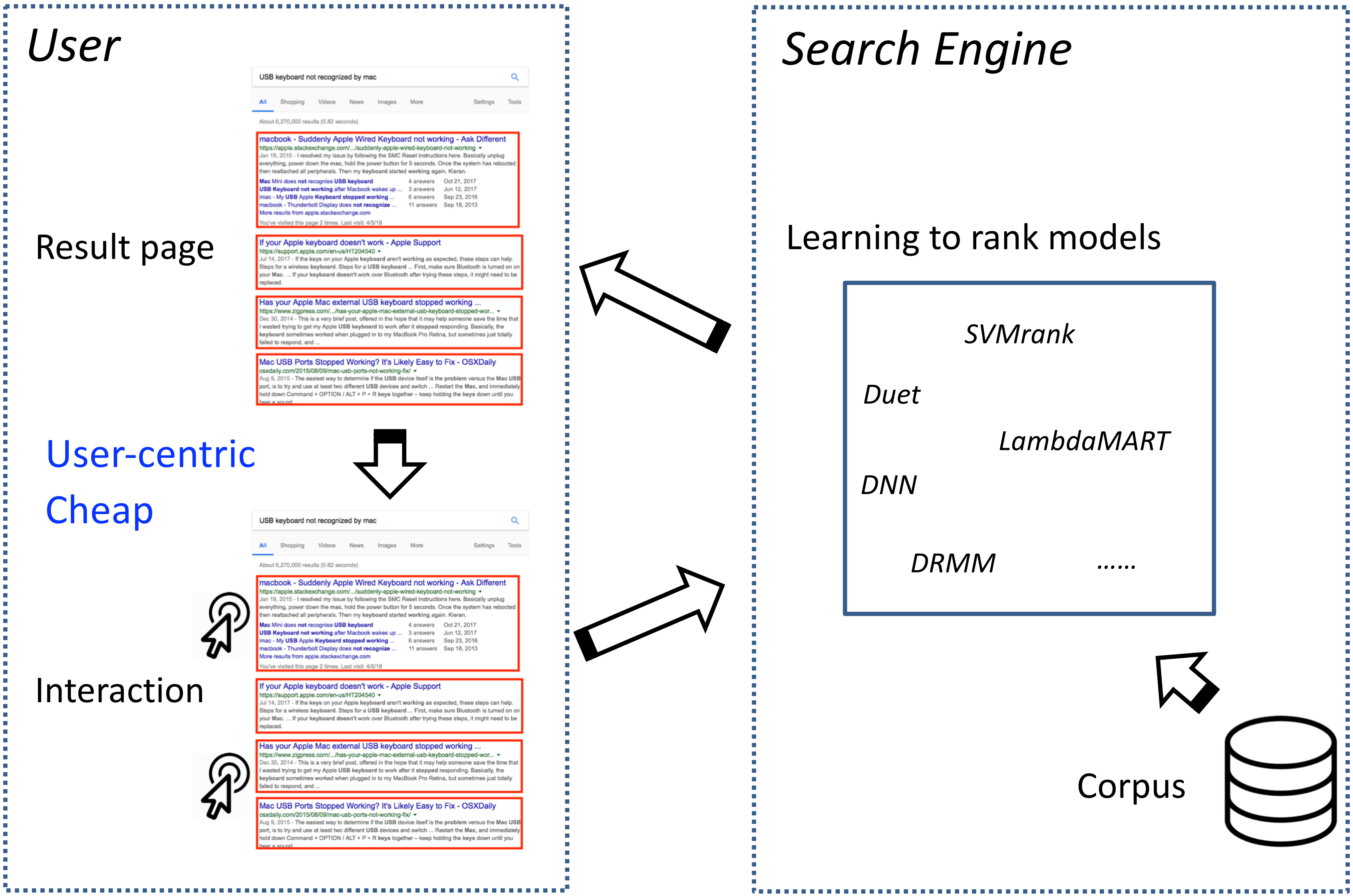
Ranking Scores

3.2
-1.4
2.8

x $S(x)$

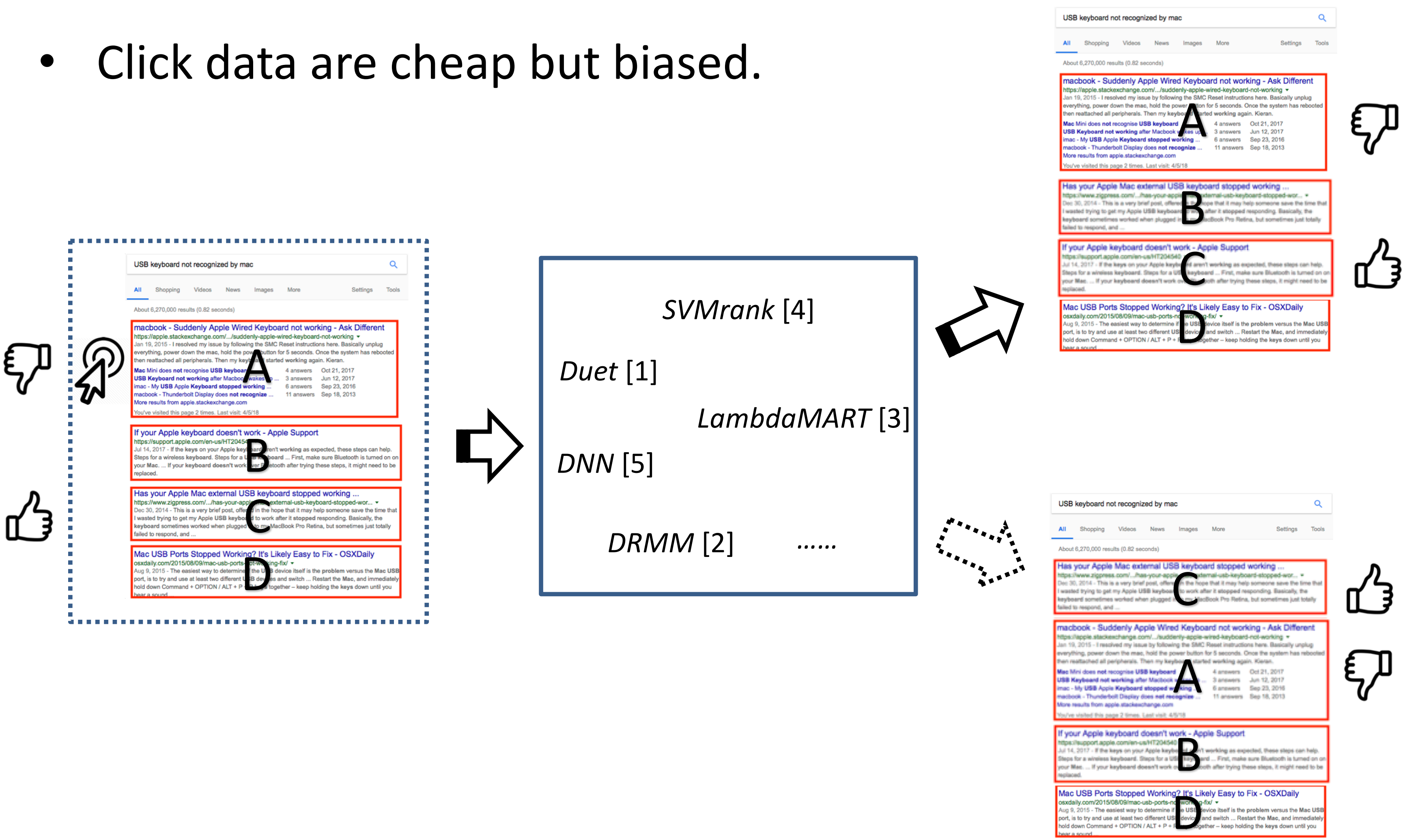
Supervision

Learning to Rank from User Feedback



Learning to Rank from User Feedback

- Click data are cheap but biased.



Problem Analysis

True ranking loss

$$\theta^* = \arg\min_{\theta} \mathcal{L}(S) = \arg\min_{\theta} \int_{q \in \mathcal{Q}} \sum_{x_i \in \pi_q, r_i = 1} \Delta(S(x_i, \theta), r_i | \pi_q)$$

X

X

Empirical ranking loss

$$\hat{\theta} = \arg\min_{\theta} \hat{\mathcal{L}}(S) = \arg\min_{\theta} \int_{q \in \mathcal{Q}} \sum_{x_i \in \pi_q, r_i = 1} \Delta(S(x_i, \theta), c_i | \pi_q)$$

x_i: ranking score

c_i: user click

r_i: true relevance label

π_q: rank list

Unbiased Learning to Rank

- Objective:

  Learning an unbiased ranker from biased user feedback.

- Solutions:
  - Calibrate biased user feedback
  - Construct unbiased learning algorithm

Solution 1: Calibrate User Feedback

True ranking loss

$$\theta^* = \arg\min_{\theta} \mathcal{L}(S) = \arg\min_{\theta} \int_{q \in \mathcal{Q}} \sum_{x_i \in \pi_q, r_i = 1} \Delta(S(x_i, \theta), r_i | \pi_q)$$

Click Model

Empirical ranking loss

$$\hat{\theta} = \arg\min_{\theta} \hat{\mathcal{L}}(S) = \arg\min_{\theta} \int_{q \in \mathcal{Q}} \sum_{x_i \in \pi_q, r_i = 1} \Delta(S(x_i, \theta), c_i | \pi_q)$$

Solution 2: Unbiased Learning Algorithm

True ranking loss

$$\theta^* = \arg\min_\theta \mathcal{L}(S) = \arg\min_\theta \int_{q\in\mathcal{Q}} \sum_{x_i\in\pi_q, r_i=1} \Delta(S(x_i,\theta), r_i|\pi_q)$$

$$\overline{\mathcal{L}}(S)$$ Unbiased Learning Algorithm

Empirical ranking loss

$$\hat{\theta} = \arg\min_\theta \hat{\mathcal{L}}(S) = \arg\min_\theta \int_{q\in\mathcal{Q}} \sum_{x_i\in\pi_q, r_i=1} \Delta(S(x_i,\theta), c_i|\pi_q)$$

Outline

- Introduction
  - Problem Analysis
  - Existing Solutions
- Part 1: Click Models
  - Basic Concept and Hypothesis
  - Advanced Click Model
  - Applications
- Part 2: Unbiased Learning Algorithms
  - Inverse Propensity Weighting
  - Online Result Randomization
  - Simulation/ Real-world Experiments
  - Advanced Topics
- Summary

Part I: Click Models

Extracting unbiased relevance feedback
from biased user clicks

Outline

- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Outline

- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Motivation

- Goal:
  - Use user click as relevance feedback to train ranking model
  - Especially useful for Web search because of a large amount of click data available

Motivation

- Problem:
  - Clicks are noisy and biased estimation of relevance:
    - $P(c_{\mathrm{x}q} = 1) \neq P(r_{\mathrm{x}q} = 1)$
    - $\mathrm{c}_{\mathrm{x}q}$: whether the user click document x for query q
    - $r_{\mathrm{x}q}$: whether document x is relevant to query q
  - For example, clicks are affected by:
    - Position bias: users tend to click high-ranked results
    - Presentation bias: users tend to click visually attractive results

Motivation

- Solution: build click models
  - Make **assumptions** about the causes of the bias
  - Construct a probabilistic **model** for the bias:
    - For click model M: $P_M(c_{xq}|r_{xq})$
    - c_{xq}: user clicks on document x for query q
    - r_{xq}: true relevance between document x and query q
  - Conduct **statistical inference** to get unbiased estimation of relevance
    - Maximum likelihood estimation
    - EM algorithm

Outline

- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Basic click models

- Including:
  - Position-based model and Cascade model (Craswell et al., 2008)
  - UBM (User Browsing Model, Dupret and Piwowarski, 2008)
  - DBN (Dynamic Bayesian Network, Chapelle and Zhang, 2009)
- Address the **Position bias**
  - i.e., users tend to click high-ranked results in Web search

Examination Hypothesis

- Why do users tend to click highly ranked results?
  - Examination bias: users are more likely to examine high-ranked results

Examination Hypothesis

- Make explicit **assumptions** about the bias
  - Examination hypothesis (Richardson et al., 2007)
    - A user clicks a document if and only if she **examined** the document and was **attracted** by the document
    - $c_i = 1 \Leftrightarrow o_i = 1 \text{ and } r_i = 1$
    - c_i: whether the user clicked the ith document
    - o_i: whether the user examined the ith document
      - Examination – reading a snippet
    - r_i: whether the user was attracted by the ith document
      - Attractiveness only depends on the snippet but can still reflect the relevance of the document
      - Also called “perceived relevance”

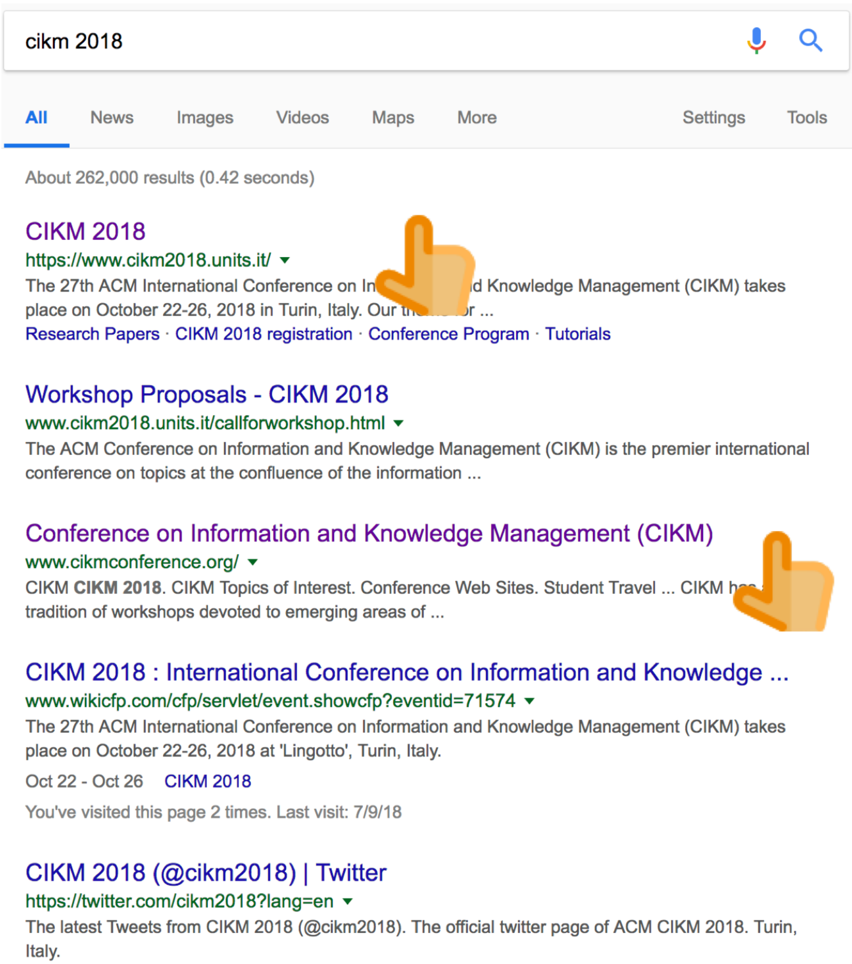
Examination Hypothesis

- Based on examination hypothesis, we can construct a **statistical model**
  - $P_M\left(c_{xq}=1\middle|\Theta\right)=P_M\left(o_{xq}=1\middle|\Theta\right)\cdot P_M\left(r_{xq}=1\middle|\Theta\right)$
  - $P_M\left(o_{xq}=1\middle|\Theta\right)$
    - Examination probability
    - Different models make different assumptions on examination patterns, and have different ways to compute the examination probability
  - $P_M\left(r_{xq}=1\middle|\Theta\right)=\alpha_{uq}$
    - α_{uq}: parameter for attractiveness or perceived relevance of document x regarding query q
    - A better (less biased) estimation of true relevance than $P(c_{xq}=1)$

Model Examination Behavior

- Different models make different assumptions on users' examination patterns, and have different implementation for $P_M(o_{xq}=1)$
- Position-based model:
  - $P_{PM}\left(o_{xq}=1\right)=\gamma_k$, where k is the rank of document x
  - The simplest click model that address position bias
  - Is used in Part 2 of this tutorial

Model Examination Behavior



$$P_{PM}(o_i = 1 | C_{1 \ldots i-1})$$

γ_1

γ_2

γ_3

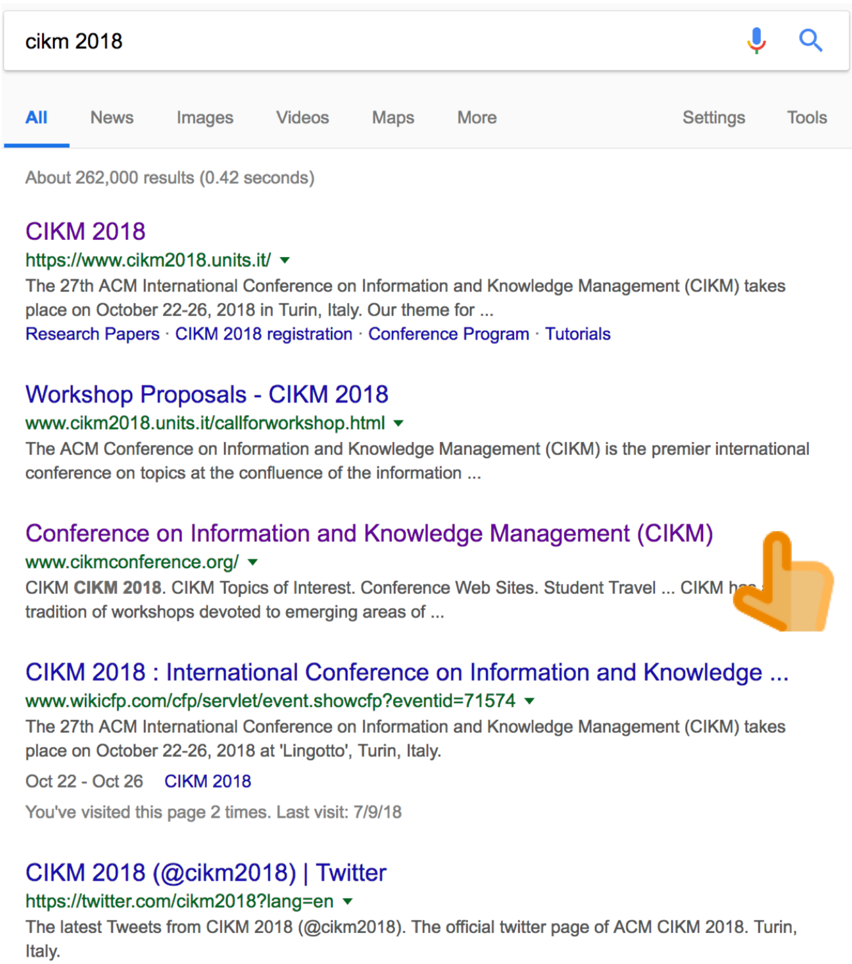
γ_4

γ_5

Model Examination Behavior

- Cascade model:
  - Assumes that a user scans documents from top to bottom until she finds and clicks a relevant document
    - $P_{CM}(o_1 = 1) = 1$
    - $P_{CM}(o_k = 1 | o_{k-1} = 0) = 0$
    - $P_{CM}(o_k = 1 | c_{k-1} = 1) = 0$
    - $P_{CM}(o_k = 1 | c_{k-1} = 0, o_{k-1} = 1) = 1$
  - Pros:
    - Simple parameter estimation (will be discussed later)
  - Cons:
    - Only models session with one click
    - The linear examination assumption is often too strong

Model Examination Behavior



$$P_{CM}(o_i = 1 | C_{1\ldots i-1})$$

1

1

1

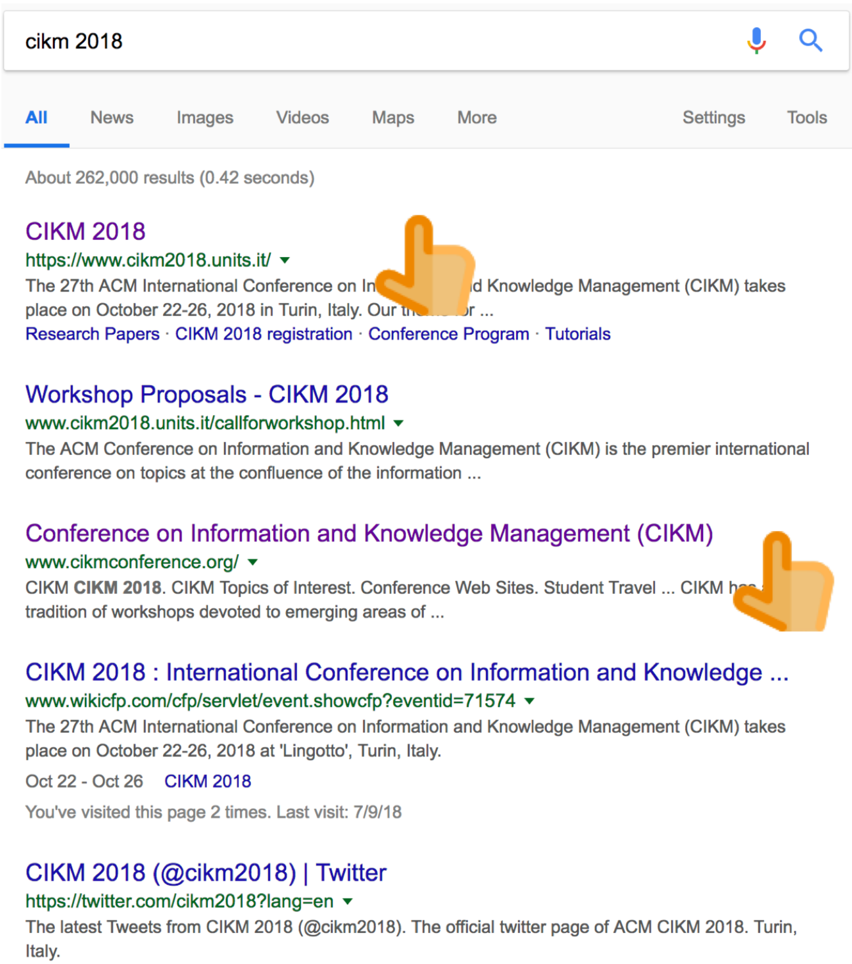
0

0

Model Examination Behavior

- User Browsing Model (UBM):
  - The examination probability depends on the rank k and the distance to last clicked document d
    - $P_{UBM}(o_k = 1) = \gamma_{kd}$
    - $d = k - \max\{j | c_j = 1, j \in \{0, \dots, k-1\}\}$
    - For convenience, we set $c_0 = 1$
  - Pros:
    - More flexible than Position-based Model and Cascade Model
    - Has a good click prediction performance

Model Examination Behavior



$$P_{UBM}(o_i = 1 | C_{1 \ldots i-1})$$

$\gamma_{1,1}$

$\gamma_{2,1}$

$\gamma_{3,2}$

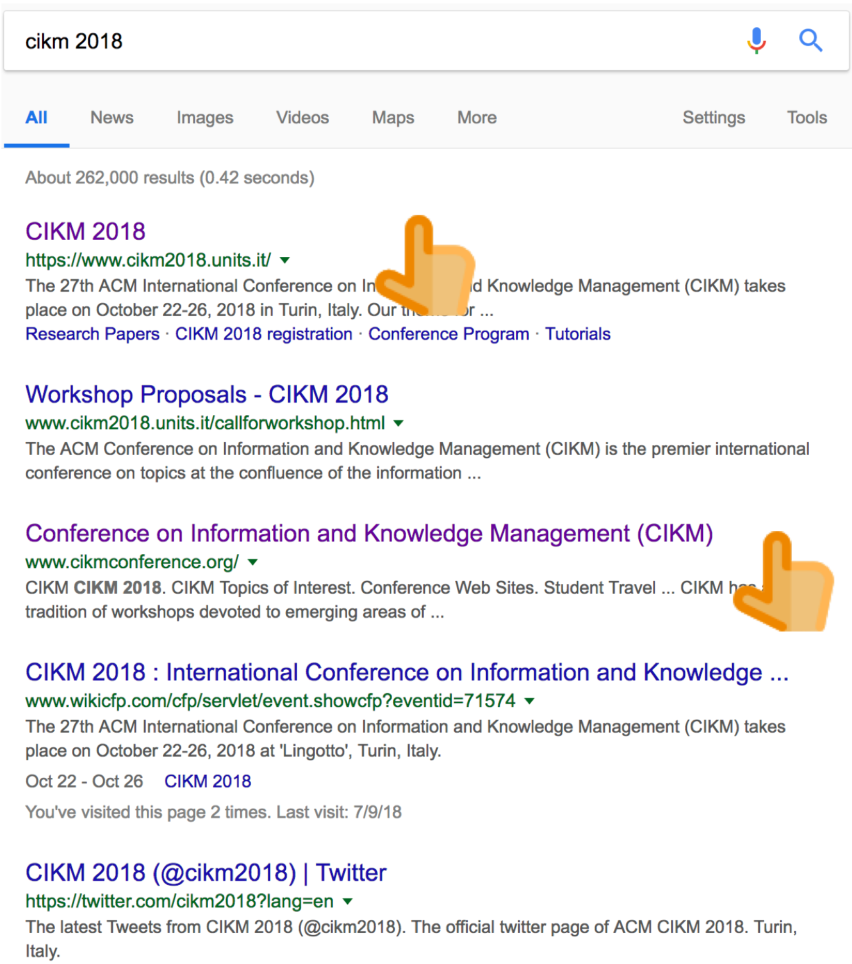
$\gamma_{4,1}$

$\gamma_{5,2}$

Model Examination Behavior

- Dynamic Bayesian Network (DBN):
  - Assumes that a user scans documents from top to bottom until she was satisfied by a clicked document
  - Use another Bernoulli variable s_k to represent whether the user is satisfied after click the document ranked at the kth position
    - $P_{DBN}(o_1 = 1) = 1$
    - $P_{DBN}(o_k = 1 | o_{k-1} = 0) = 0$
    - $P_{DBN}(s_k = 1 | c_k = 1) = \sigma_{x_k q}$
    - $P_{CM}(o_k = 1 | s_{k-1} = 1) = 0$
    - $P_{CM}(o_k = 1 | s_{k-1} = 0, o_{k-1} = 1) = \gamma$
  - Pros:
    - Not only models the attractiveness with $\alpha_{x_k q}$ but also the intrinsic relevance (i.e., the relevance of clicked document) with $\sigma_{x_k q}$
    - $rel_{xq} \propto \alpha_{xq} \cdot \sigma_{xq}$

Model Examination Behavior



$$P_{DBN}(o_i = 1|C_{1\dots i-1})$$

$$1$$

$$\frac{P(o_1 = 1)\times\alpha_i(1-\sigma_1)\gamma}{P(o_1 = 1)\times\alpha_i} = (1-\sigma_1)\gamma$$

$$\frac{P(o_2 = 1|c_1)\times(1-\alpha_2)\gamma}{1-P(o_2 = 1|c_1)\times\alpha_2}$$

$$(1-\sigma_3)\gamma$$

$$\frac{P(o_4 = 1|C_{1\dots3})\times(1-\alpha_4)\gamma}{1-P(o_4 = 1|C_{1\dots3})\times\alpha_4}$$

Inference for Model Parameters

- Click likelihood on training set:

$$LL(S,\Theta) = \sum_{s\in S} \log P(c_1^{(s)}, c_2^{(s)}, \dots, c_{M_s}^{(s)} | \Theta)$$

  - S: search sessions in the training set
  - $c_i^{(s)}$: whether the user clicked the ith document in session s
  - M_s: the number of documents displayed in session s, usually $M_s = 10$
  - Θ: parameters of the click model that need to learn. For example, for UBM, $\Theta = (\{\gamma_{kd}\},\ \{\alpha_{xq}\})$

- Maximum likelihood estimation:

$$\widehat{\Theta} = \mathrm{argmax}_{\Theta}\, LL(S,\Theta)$$

Inference for Model Parameters

- MLE for Cascade Model:
  - For session s, there is a click on rank l_s.
  - According to model assumptions, we have:
    - $o_1 = o_2 = \cdots = o_{l_s} = 1$
    - $o_{l_s+1} = o_{l_s+2} = \cdots = o_{M_s} = 0$
    - $r_{l_s} = 1$
    - $r_1 = r_2 = \cdots r_{l_s-1} = 0$
  - Therefore:
    - $LL_{CM}(\mathrm{S}, \Theta) = \sum_{s \in S} \log \prod_{k=1}^{l_s-1} (1 - \alpha_{x_k q_s}) \cdot \alpha_{x_{l_s} q_s}$
      $= \sum_{s \in S} \sum_{k=1}^{l_s} [I(k < l_s) \log(1 - \alpha_{x_k q_s}) + I(k = l_i) \log(\alpha_{x_k q_s})]$

Inference for Model Parameters

- Group the formula w.r.t. α_{xq}
  - $LL_{CM}(\alpha_{xq}) =$
    $$\sum_{s \in S_{xq}} \sum_{k=1}^{l_s} I(k < l_s, x_k = x, q_s = q) \log(1 - \alpha_{xq}) +$$
    $$\sum_{s \in S_{xq}} \sum_{k=1}^{l_s} I(k = l_s, x_k = x, q_s = q) \log(\alpha_{xq}) + \text{Constant}$$
- Take a partial derivative w.r.t. α_{xq} and let it equals to 0
  - $$\widehat{\alpha_{xq}} = \frac{\sum_{s \in S_{xq}} \sum_{k=1}^{l_s} I(k=l_s, x_k=x, q_s=q)}{\sum_{s \in S_{xq}} \sum_{k=1}^{l_s} I(k=l_s, x_k=x, q_s=q) + \sum_{s \in S_{xq}} \sum_{k=1}^{l_s} I(k<l_s, x_k=x, q_s=q)}$$

Inference for Model Parameters

- MLE for UBM:

$$LL_{UBM}(S,\Theta) = \sum_{s \in S} \log P\left(c_1^{(s)}, c_2^{(s)}, \dots, c_{M_s}^{(s)} \middle| \Theta\right)$$

$$= \sum_{s \in S} \log \sum_{O^{(s)}, R^{(s)}} P(C^{(s)}, O^{(s)}, R^{(s)} | \Theta)$$

  - $\{O^{(s)}\}_{s \in S}$ and $\{R^{(s)}\}_{s \in S}$ are **latent variables** indicating whether users examined a document and whether the document was attractive to the user
  - Difficult to enumerate all possible $O^{(s)}, R^{(s)}$ and compute LL_{UBM}
    - **Use Expectation-Maximization algorithm**

Inference for Model Parameters

- EM for UBM:

$$LL_{UBM}(S,\Theta)=\sum_{s\in S}\log\sum_{O^{(s)},R^{(s)}}P(\,C^{(s)},O^{(s)},R^{(s)}|\Theta)$$

$$Q_{UBM}(\Theta|\Theta')=\sum_{s\in S}E_{O^{(s)},R^{(s)}|C^{(s)},\Theta'}\log P(\,C^{(s)},O^{(s)},R^{(s)}|\Theta)$$

- E-step:
  - Compute $P\left(O^{(s)},R^{(s)}\middle|C^{(s)},\Theta'\right)$ with old parameters Θ'
- M-step:
  - Maximize $Q_{UBM}(\Theta|\Theta')$ w.r.t. Θ to update new parameters Θ
- Iteratively take E-step and M-step until convergence

Inference for Model Parameters

- EM for UBM:

$$
\begin{aligned}
Q_{UBM}(\Theta|\Theta') &= \sum_{s\in S} E_{O^{(s)},R^{(s)}|C^{(s))},\Theta'} \log P(C^{(s))}, O^{(s)}, R^{(s)}|\Theta) \\
&= \sum_{s\in S} E_{O^{(s)},R^{(s)}|C^{(s)}} \\
&\quad \sum_{k=1}^{M_s} \log[P(c_k^{(s)}|o_k^{(s)}, r_k^{(s)})P(o_k^{(s)}|C_{<k}^{(s)}, \gamma_{rd})P(r_k^{(s)}|\alpha_{x_k q_s})]
\end{aligned}
$$

Inference for Model Parameters

- Group the formula w.r.t. α_{xq}

$$
\begin{aligned}
Q(\alpha_{xq}|\Theta') = \sum_{s \in S_{xq}} & I(C_x^{(s)} = 1) P(r_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') \log(\alpha_{xq}) \\
& + I(c_x^{(s)} = 0) P(r_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') \log(\alpha_{xq}) \\
& + I(c_x^{(s)} = 0) P(r_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') \log(1 - \alpha_{xq}) \\
& + \text{Constant}
\end{aligned}
$$

  - S_{xq}: search sessions initiated by query q and containing document x
  - $c_x^{(s)}$: observed click on document u in session s
  - $r_x^{(s)}$: latent variable indicating whether document u is attractive in in session s

Inference for Model Parameters

- Group the formula w.r.t. α_{xq}

$$\begin{aligned} Q(\alpha_{xq}|\Theta') = \sum_{s \in S_{xq}} & I(C_x^{(s)} = 1) P(r_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') \log(\alpha_{xq}) \\ & + I(c_x^{(s)} = 0) P(r_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') \log(\alpha_{xq}) \\ & + I(c_x^{(s)} = 0) P(r_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') \log(1 - \alpha_{xq}) \\ & + \text{Constant} \end{aligned}$$

- In E-step, compute:

$$P(r_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') = 1$$

$$P(r_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') = \frac{\alpha'_{xq}(1 - \gamma'_{rd})}{1 - \alpha'_{xq} \cdot \gamma'_{rd}}$$

$$P(r_x^{(s)} = 0 | C_x^{(s)} = 0, \Theta') = \frac{1 - \alpha'_{xq}}{1 - \alpha'_{xq} \cdot \gamma'_{rd}}$$

Inference for Model Parameters

- Group the formula w.r.t. α_{xq}

$$
\begin{aligned}
Q(\alpha_{xq}|\Theta') = \sum_{s \in S_{xq}} & I(C_x^{(s)} = 1) P(r_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') \log(\alpha_{xq}) \\
& + I(c_x^{(s)} = 0) P(r_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') \log(\alpha_{xq}) \\
& + I(c_x^{(s)} = 0) P(r_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') \log(1 - \alpha_{xq}) \\
& + \text{Constant}
\end{aligned}
$$

- In M-step, maximize $Q(\alpha_{xq}|\Theta')$

$$
\frac{\partial Q(\alpha_{xq}|\Theta')}{\partial \alpha_{xq}} = 0
$$

$$
\alpha_{xq} = \frac{1}{|S_{xq}|} \sum_{s \in S_{xq}} [c_x^{(s)} + (1 - c_x^{(s)}) \frac{\alpha'_{xq}(1 - \gamma'_{rd})}{1 - \alpha'_{xq} \cdot \gamma'_{rd}}]
$$

Inference for Model Parameters

- Group the formula w.r.t. γ_{rd}

$$
\begin{aligned}
Q(\gamma_{rd}|\Theta') = \sum_{s \in S_{rd}} & I(c_x^{(s)} = 1) P(o_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') \log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0) P(o_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') \log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0) P(o_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') \log(1 - \gamma_{rd}) \\
& + \text{Constant}
\end{aligned}
$$

  - S_{rd}: search sessions that with a click at rank r and a previous click at rank $r - d$
  - $c_{\mathrm{x}}^{(s)}$: observed click on document x in session s
  - $o_x^{(s)}$: latent variable indicating whether document x is examined in in session s
  - r: rank of document x
  - d: distance from document x to the last click document ranked higher than x

Inference for Model Parameters

- Group the formula w.r.t. γ_{rd}

$$
\begin{aligned}
Q(\gamma_{rd}|\Theta') = \sum_{s \in S_{rd}} & I(c_x^{(s)} = 1) P(o_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') \log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0) P(o_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') \log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0) P(o_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') \log(1 - \gamma_{rd}) \\
& + \text{Constant}
\end{aligned}
$$

- In E-step, compute:

$$
\begin{aligned}
P(o_x^{(s)} = 1 | c_x^{(s)} = 1, \Theta') &= 1 \\
P(o_x^{(s)} = 1 | c_x^{(s)} = 0, \Theta') &= \frac{(1 - \alpha'_{xq})\gamma'_{rd}}{1 - \alpha'_{xq} \cdot \gamma'_{rd}} \\
P(o_x^{(s)} = 0 | c_x^{(s)} = 0, \Theta') &= \frac{1 - \gamma'_{rd}}{1 - \alpha'_{xq} \cdot \gamma'_{rd}}
\end{aligned}
$$

Inference for Model Parameters

- Group the formula w.r.t. γ_{rd}

$$
\begin{aligned}
Q(\gamma_{rd}|\Theta') = \sum_{s \in S_{rd}} & I(c_x^{(s)} = 1)P(o_x^{(s)} = 1|c_x^{(s)} = 1, \Theta')\log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0)P(o_x^{(s)} = 1|c_x^{(s)} = 0, \Theta')\log(\gamma_{rd}) \\
& + I(c_x^{(s)} = 0)P(o_x^{(s)} = 0|c_x^{(s)} = 0, \Theta')\log(1 - \gamma_{rd}) \\
& + \text{Constant}
\end{aligned}
$$

- In M-step, maximize $Q(\gamma_{rd}|\Theta')$

$$
\frac{\partial Q(\gamma_{rd}|\Theta')}{\partial \gamma_{rd}} = 0
$$

$$
\gamma_{rd} = \frac{1}{|S_{rd}|}\sum_{s \in S_{rd}} [c_x^{(s)} + (1 - c_x^{(s)})\frac{(1 - \alpha'_{xq})\gamma'_{rd}}{1 - \alpha'_{xq} \cdot \gamma'_{rd}}]
$$

Inference for Model Parameters

- EM for DBN
  - Sequential dependency between latent variables o_k, s_{k-1}, and o_{k-1}
- In E-step:
  - Use forward-backward algorithm to estimate $P(r_k|C)$ and $P(s_k|C)$
  - See the appendix of Chapelle and Zhang (2009) for more details

Basic click models: summary

- Address the **position bias**
- Make **assumptions** about the causes of the bias
  - Examination hypothesis
  - Different examination probabilities at different positions
- Construct a probabilistic **model** for the bias
  - Position-based model
  - Cascade model
  - UBM
  - DBN
- Conduct **statistical inference** to get unbiased estimation of relevance
  - MLE
  - EM algorithm

Outline

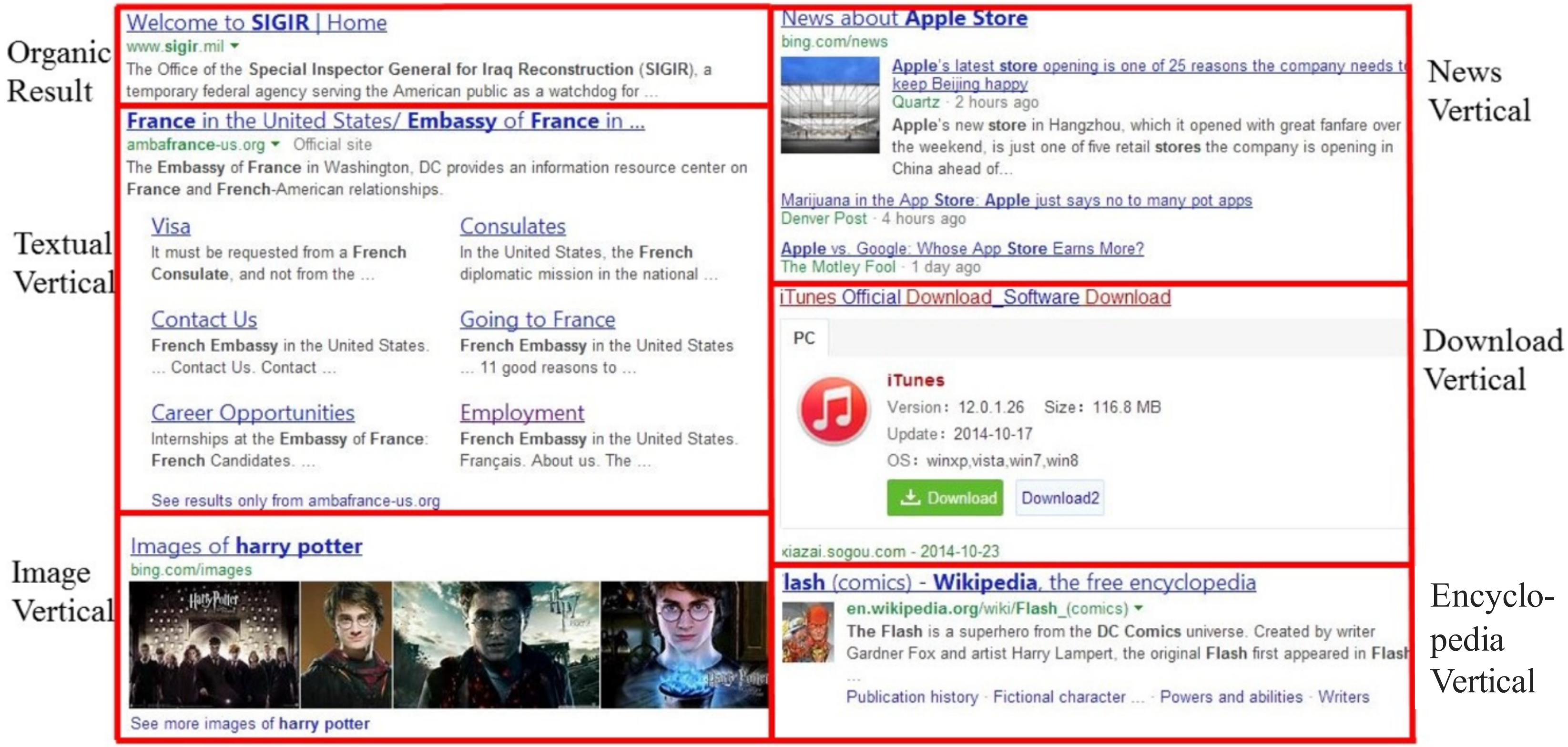
- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Advanced click models

- Model other behavior biases
  - Vertical-aware click model (VCM, Wang et al., 2013)
    - Models the presentation bias of different vertical results
  - Mobile click model (MCM, Mao et al., 2018)
    - Models the click necessity bias and examination satisfaction in mobile search
- Exploit diverse behavior signals
  - Time-aware click model (TACM, Liu et al., 2016)
    - Incorporate dwell time into click models
  - Enhance click models with mouse movement data (Liu et al., 2017)
    - Use mouse movement features to improve the estimation of examination probability

Vertical-aware click model

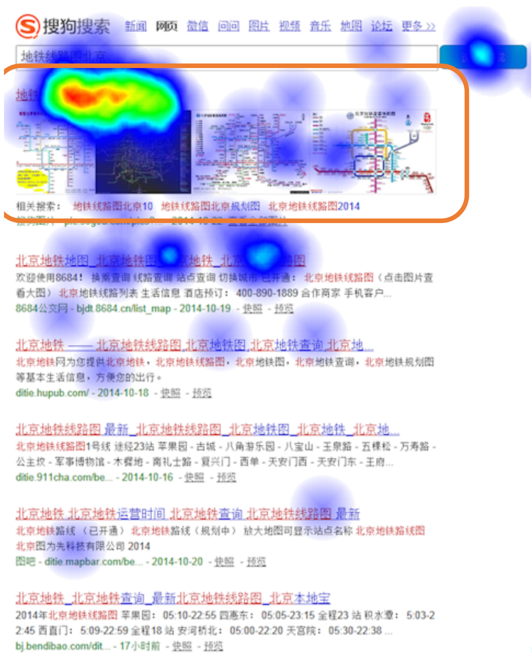
- The SERPs are becoming heterogenous
  - With different types of vertical results



Vertical-aware click model

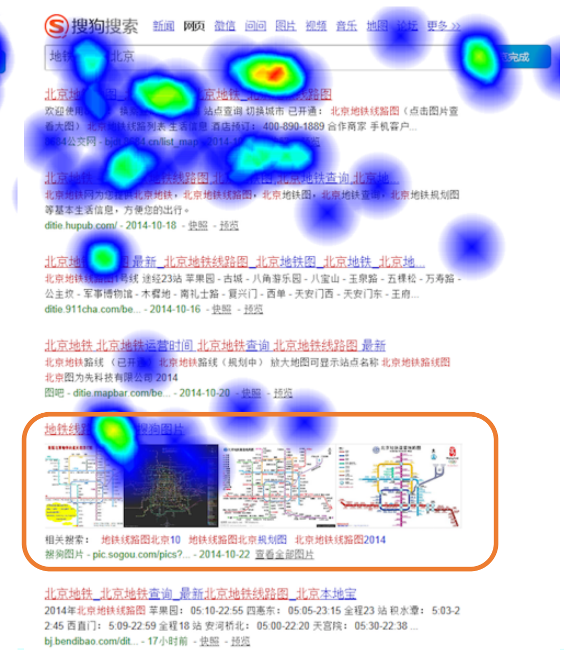
- Presentation bias
  - Certain types of vertical results draw much attention from users
  - Users may examine a vertical result first

Rank 1st



Rank 3rd

Rank 5th



Vertical-aware click model

- Restart effect
  - Most users (~70%) will restart from the beginning after examining a vertical results

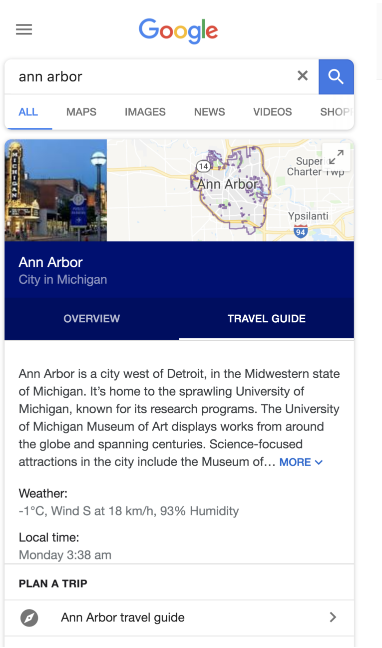
Vertical-aware click model

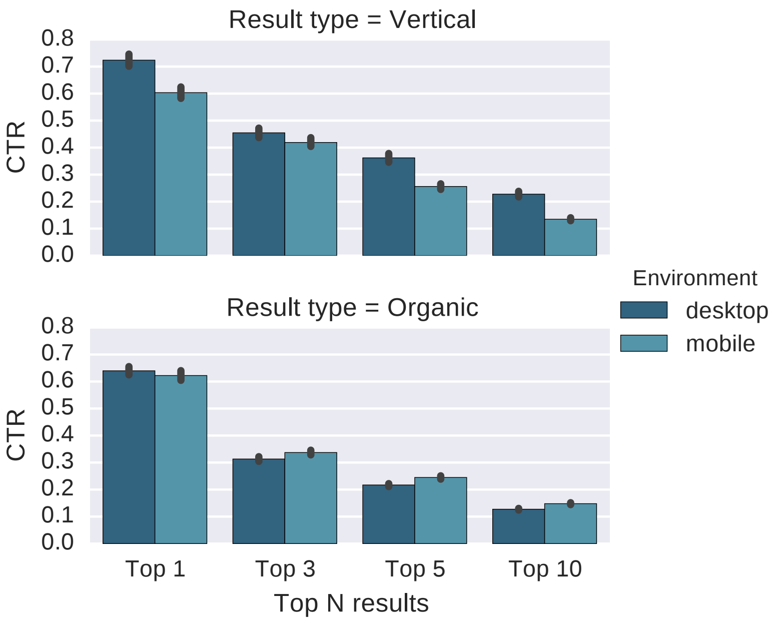
- Vertical-aware click model
  - f: whether the user examines a vertical result first
  - b: whether the user restart from the beginning after examining the vertical result
  - t_v: the type of the vertical result
  - l_v: the rank of the vertical result

Original UBM	$P(c_i = 1 \mid o_i = 1) = 0$ $P(c_i = 1 \mid o_i = 1) = P(r_i = 1 \mid o_i = 1)$ $P(o_i = 1 \mid f = 0, C_{1:i-1}) = \gamma_{rd}$ $P(r_i = 1 \mid o_i = 1, f = 0) = \alpha_{xq}$
Users examine vertical results at first	$P(f = 1) = \phi_{t_v, l_v}$
Effect on Examination	$P(o_i = 1 \mid f = 1, C_{1:i-1}) = \gamma_{rd} + \theta_{q,i}$
Effect on Click-through	$P(r_i = 1 \mid o_i = 1, f = 1) = \alpha_{q,i} + \beta_{q,i}$
Restart effect	$P(b = 1 \mid f = 0) = 0$ $P(b = 1 \mid f = 1) = \sigma_{t_v, l_v}$

Mobile click model

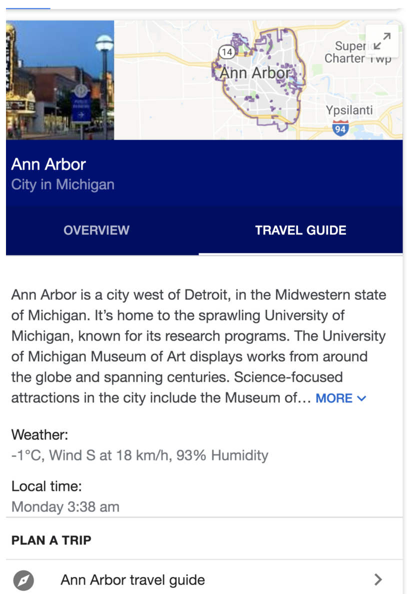
- Mobile search is different from desktop search
  - For example: some mobile search results are designed to satisfy users' information needs without click
  - Vertical results with low click necessity => lower CTR





Mobile click model

- Click necessity bias:
  - Some types of search results have low click necessity because they can satisfy users' information needs without requiring any clicks, which will lower the click probabilities of these results
  - Avoid negative feedback on results with lower click necessity and CTR

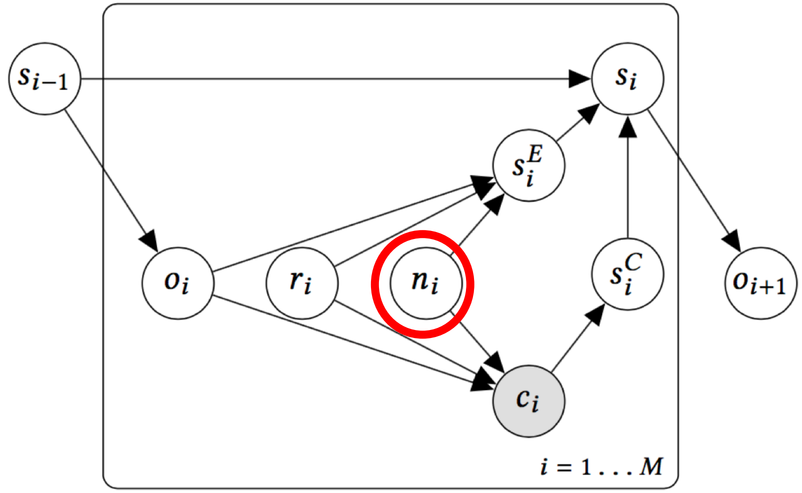


Mobile click model

- Model click necessity bias:
  - Extending Examination hypothesis:

$$c_i = 1 \Leftrightarrow o_i = 1 \land r_i = 1 \land n_i = 1$$
$$P(n_i = 1) = \beta_{v_i}$$

  - n_i: the click necessity of the ith document, only depends on the type of the search result v_i

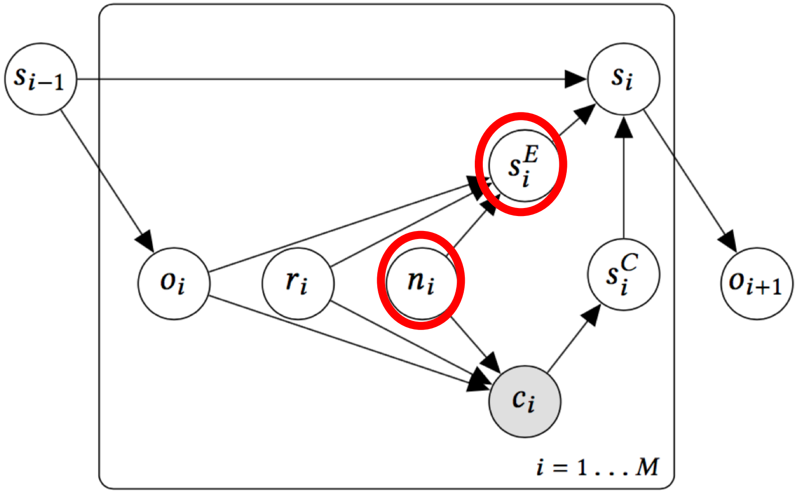


Mobile click model

- Examination satisfaction bias:
  - A user can feel satisfied and leave the SERP after examining a search result that is both attractive and with low click necessity.
  - Give positive feedback for the results with low click necessity

Mobile click model

- Model examination satisfaction bias :
  - Use s_i to denote user's state of satisfaction after position i:
  - $s_i = 1 \Leftrightarrow s_{i-1} = 1 \vee (s_i^e = 1 \vee s_i^c = 1)$
  - $P(s_i^c = 1 | c_i = 1) = \sigma_{q,d_i}^c$ (Click satisfaction)
  - $P(s_i^e = 1 | o_i = 1, r_i = 1, n_i = 0) = \sigma_{q,d_i}^e$ (Examination satisfaction)



Mobile click model

- Mobile click model can infer the parameters for click necessity and examination satisfaction probability
  - Click necessity parameters: β
  - Examination satisfaction probability: s^E

Query: 双色球开奖结果 (The result of “双色球”, a popular lottery in China)	**Query: wifi查看密码 (viewing wifi password)**	**Query: 焦虑症 (Anxiety disorder)**
Result: A direct answer result showing the result of the latest lottery and the payment of the prizes.	Result: A vertical result that provides step-by-step guidance of how to view the wifi password on a PC.	Result: An medical knowledge graph result showing the symptom, diagnose, and treatment of Anxiety disorder on the SERP.
$\beta = 0.0001, s^E = 0.904$	$\beta = 0.0002, s^E = 0.5$	$\beta = 0.0003, s^E = 0.322$

Time-aware click model

- Incorporate dwell time into click models
  - Click dwell time can be extracted from query logs
  - Assumption: Longer click dwell time indicate a higher probability of satisfaction
  - Model:

    $$P(s_i = 1) = P(r_i = 1) \times F(DwellTime_i)$$
    $$F \in R^+ \to [0, 1]$$

  - s_i: whether the user is satisfied after clicking the ith document
  - F: Function that maps dwell time to a probability
    - Linear, exponential,…

Time-aware click model

- Incorporate dwell time into click models
  - Improve the performance of relevance estimation

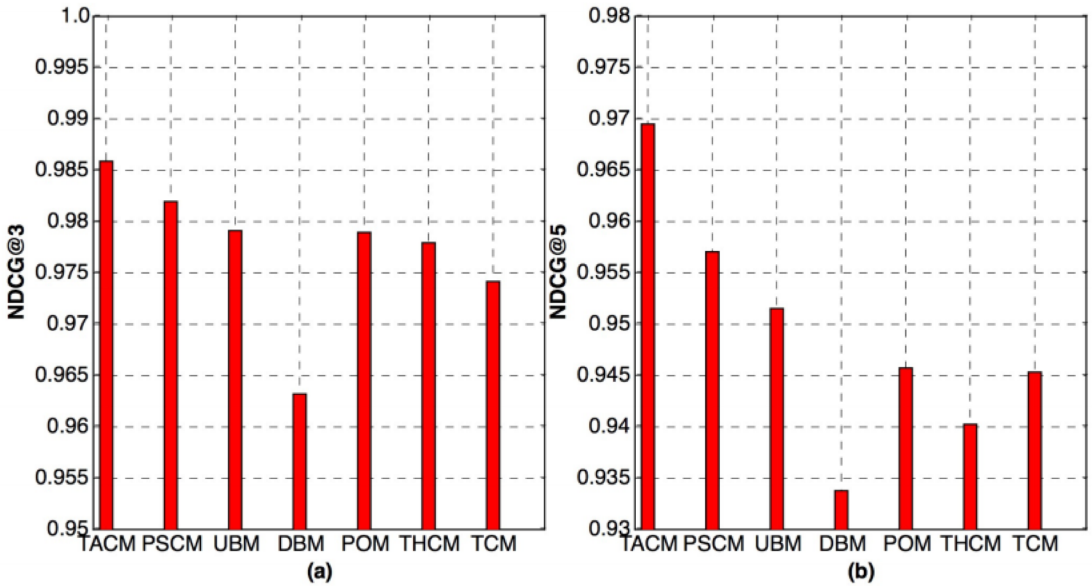


Fig. 5. Relevance estimation performance in terms of NDCG@3 and NDCG@5 for Data-C (All differences are statistically significant ($p - value < 0.05$) according to paired t-test).

Mouse-enhanced click model

- Enhance click models with mouse movement data
  - Mouse movement on SERPs can be logged by javascripts
  - Assumption: mouse movement can reflect user's attention

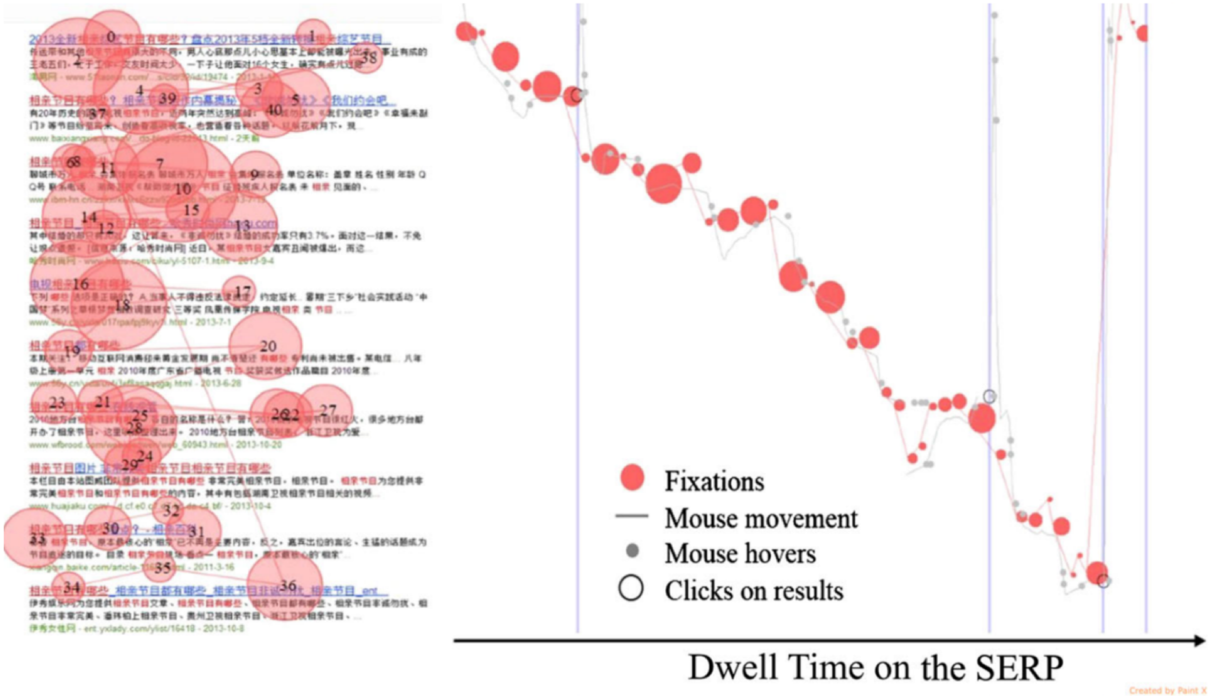


Fig. 1 Mouse/eye tracking results on a SERP for a user after submitting a query "TV Dating shows"

Mouse-enhanced click model

- Enhance click models with mouse movement data
  - Model:
    - Use mouse movement features to predict whether the user examined a result

Table 1 Description of features extracted from mouse movement data

Feature name	Description
MostLeft	The most left position that cursor reaches in the result's display area
HorizontalMoveRight	The total rightward distance of cursor in the result's display area
DwellTime	The total dwell duration that cursor stays in the result's display area
VerticalDwellTime	The total dwell duration that cursor stays within the result's display area horizontally
HoverTime	The total duration that cursor hovers over the result's display area
ActionNumber	The number of cursor actions that happen in the result's display area

Mouse-enhanced click model

- Enhance click models with mouse movement data
  - Model:
    - Use mouse movement features to predict the probability that the user examined a result $P(m_i = 1)$
    - Linearly combine the mouse movement based examination prediction with $P(o_i = 1)$
    - For example, for UBMwM:
    - $P(o_i = 1) = (1 - w)\gamma_{id} + wP(m_i = 1)$

Mouse-enhanced click model

- Enhance click models with mouse movement data
  - Mouse movement data is effective in estimating user attention
  - Enhanced click models has a better performance in relevance estimation task

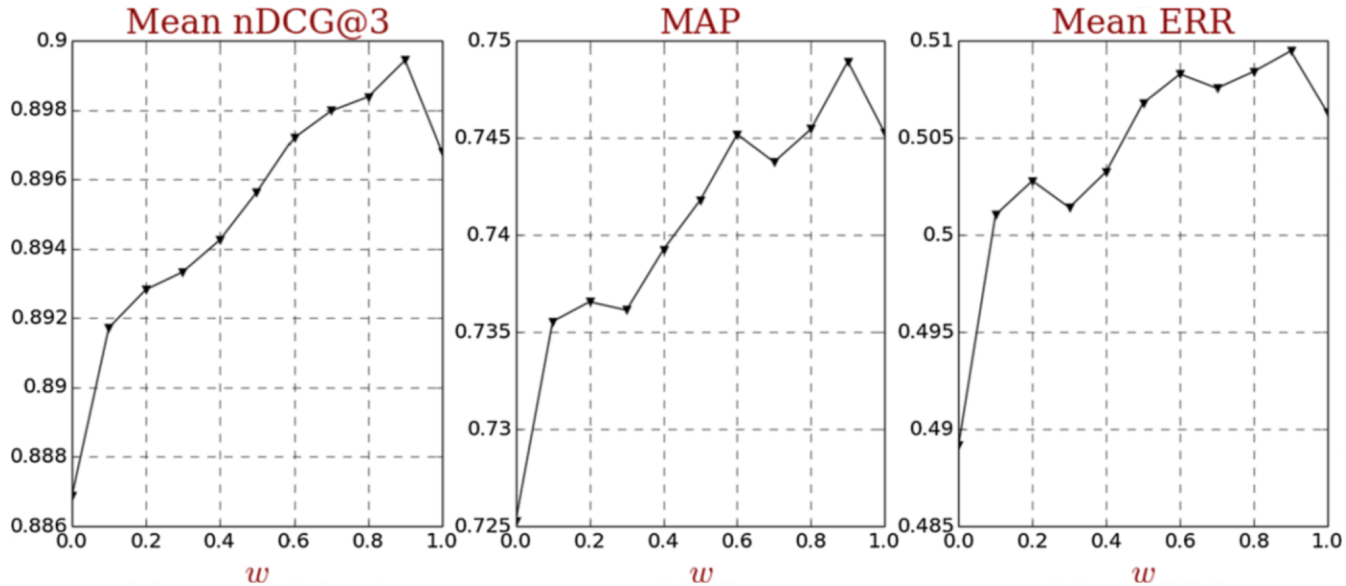


Fig. 7 Relevance prediction results of UBMwM with different w ($w = 0$ means original UBM)

Outline

- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Application: re-rank documents

- Infer unbiased relevance estimation for documents appeared in query logs
  - For example, α_{xq} for position-based model, cascade model, and UBM; $\alpha_{xq} \cdot \sigma_{xq}$ for DBN
- Re-rank the documents according to their relevance parameters
- Or use relevance estimation as ranking features
  - Gap between click relevance and relevance annotation
- Problem: only work for query and documents appeared in query logs

Application: train learning to rank models

- Infer unbiased relevance estimation for documents appeared in query logs
- Use the unbiased relevance estimations as (weak) supervision signal to train learning to rank model
- The trained learning to rank model can generalize to unseen documents
- Useful for training deep ranking models
  - (Xiong et al., 2017)
  - (Luo et al., 2017)

Outline

- Motivation
  - Get unbiased relevance feedback from biased user click
- Basic click models
  - Examination hypothesis
  - Model examination behavior
  - Statistical inference for model parameters
- Advanced click models
  - Model other behavior biases
  - Exploit diverse behavior signals
- Application of click models
  - Re-rank documents
  - Train learning to rank models
- Summary

Summary

- Goal:
  - extract unbiased relevance feedback from biased user clicks
- Approach: construct click models
  - Make assumptions on user behavior
  - Construct probabilistic models
  - Infer and estimate the unbiased relevance parameters
- Pros:
  - Simple and efficient
  - Flexible framework
- Cons:
  - Only work for the queries and documents that appeared in logs
  - Should be combined with ranking models (Part II)

Part 1

Q&A

aiqy@cs.umass.edu, maojiaxin@gmail.com

Outline

- Introduction
  - Problem Analysis
  - Existing Solutions
- Part 1: Click Models
  - Basic Concept and Hypothesis
  - Advanced Click Model
  - Applications
- Part 2: Unbiased Learning Algorithms
  - Inverse Propensity Weighting
  - Online Result Randomization
  - Simulation/ Real-world Experiments
  - Advanced Topics
- Summary

Part 2:

Unbiased Learning Algorithm

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - Dual Learning Algorithm

Motivation

True ranking loss

$$\theta^* = \arg\min_\theta \mathcal{L}(S) = \arg\min_\theta \int_{q\in\mathcal{Q}} \sum_{x_i\in\pi_q, r_i=1} \Delta(S(x_i,\theta), r_i|\pi_q)$$

$$\overline{\mathcal{L}}(S)$$

?

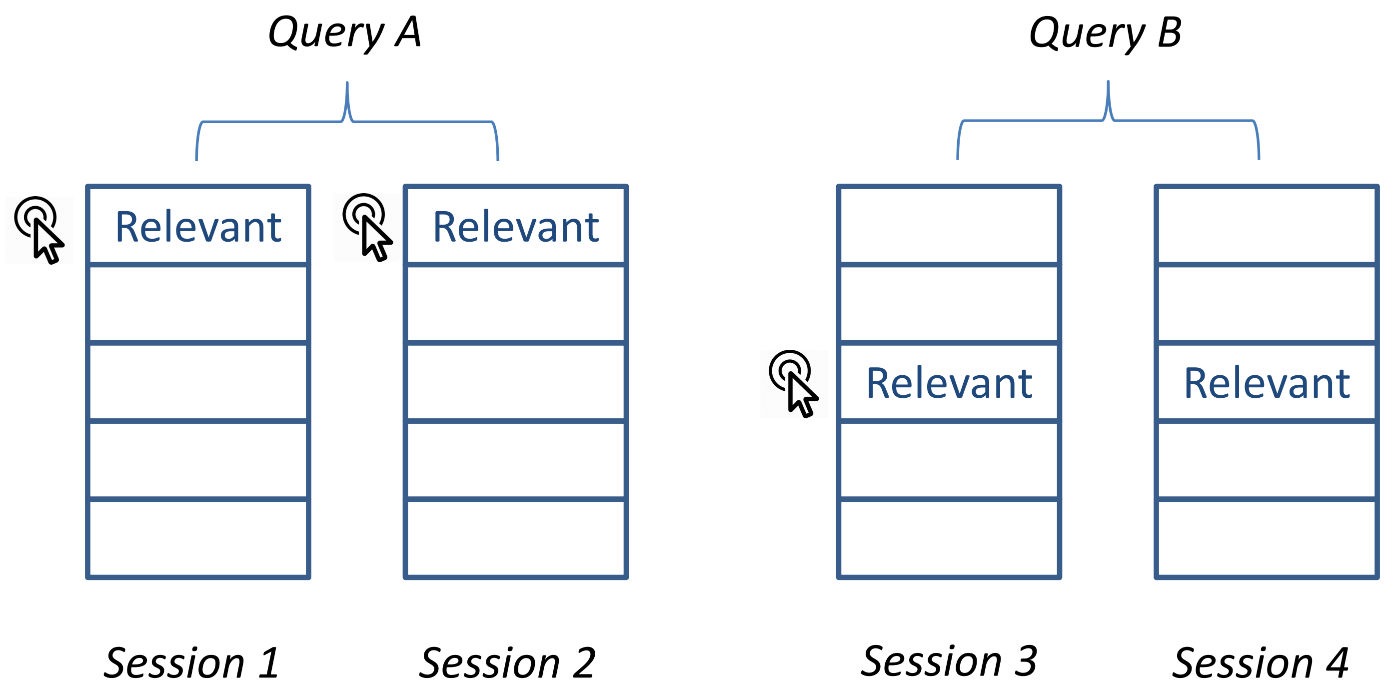
X

Empirical ranking loss

$$\hat{\theta} = \arg\min_\theta \hat{\mathcal{L}}(S) = \arg\min_\theta \int_{q\in\mathcal{Q}} \sum_{x_i\in\pi_q, r_i=1} \Delta(S(x_i,\theta), c_i|\pi_q)$$

Selection Bias

- Query-document pairs on different positions have different probability to be selected for training.



Wang et al. SIGIR 2016

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - Dual Learning Algorithm

Problem Analysis

 $=$ 

and 

$$P(c_i = 1) = P(o_i = 1) \bullet P(r_i = 1)$$

$$P(c_i = 1) \ / \ P(o_i = 1) \Rightarrow P(r_i = 1)$$

Wang et al. SIGIR 2016

Inverse Propensity Weighting

- Weight each training instance with its inverse examination propensity.

$$\hat{l}(S,q) = \sum_{x_i \in \pi_q, c_i=1} \Delta(x_i, c_i | \pi_q)$$

$$l_{IPW}(S,q) = \sum_{x_i \in \pi_q, c_i=1} \frac{\Delta(x_i, c_i | \pi_q)}{P(o_i = 1 | \pi_q)}$$

x_i: ranking score $f(d_i, \theta)$.

c_i: user click.

r_i: true relevance label.

π_q: rank list.

Wang et al. SIGIR 2016

Proof of Correctness

$$\mathbb{E}_{\mathbf{o}_q}[l_{IPW}(S,q)] = \boxed{\mathbb{E}_{\mathbf{o}_q}\Big[\sum_{x_i\in\pi_q,o_i=1,r_i=1}\frac{\Delta(x_i,r_i|\pi_q)}{P(o_i=1|\pi_q)}\Big]}$$

$c_i{=}1 \Rightarrow o_i{=}1, r_i{=}1$

$$= \mathbb{E}_{\mathbf{o}_q}\Big[\sum_{x_i\in\pi_q,r_i=1}\frac{o_i\cdot\Delta(x_i,r_i|\pi_q)}{P(o_i=1|\pi_q)}\Big]$$

$$= \sum_{x_i\in\pi_q,r_i=1}\mathbb{E}_{\mathbf{o}_q}[o_i]\cdot\frac{\Delta(x_i,r_i|\pi_q)}{P(o_i=1|\pi_q)}$$

$$= \sum_{x_i\in\pi_q,r_i=1}P(o_i=1|\pi_q)\cdot\frac{\Delta(x_i,r_i|\pi_q)}{P(o_i=1|\pi_q)}$$

$$\boxed{= \sum_{x_i\in\pi_q,r_i=1}\Delta(x_i,r_i|\pi_q)}$$

Bias removed!

Joachims et al. WSDM 2017

Proof of Robustness

- Suppose that clicks are not noise-free:

$$P(c_i = 1 | r_i = 1, o_i = 1) = \epsilon_+$$

$$P(c_i = 1 | r_i = 0, o_i = 1) = \epsilon_- \qquad \epsilon_+ > \epsilon_-$$

- Suppose that the loss of each document only depends on its relevance and position:

$$l(S, q) = \sum_{x_i \in \pi_q, r_i = 1} \Delta(x_i, r_i | \pi_q) = \sum_{x_i \in \pi_q, r_i = 1} f(rank(x_i | S)) \cdot r_i$$

Joachims et al. WSDM 2017

Proof of Robustness

$$\boxed{\mathbb{E}[l_{IPW}(S_1, q)] > \mathbb{E}[l_{IPW}(S_2, q)]}$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q, \pi_q}\left[\sum_{x_i \in \pi_q, c_i = 1} \frac{f(rank(x_i|S_1)) - f(rank(x_i|S_2))}{P(o_i = 1|\pi_q)}\right] > 0$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q}\left[\sum_{x \in \pi_q} P(c_x = 1|o_x = 1, r_x)\delta f(x)\right] > 0$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q}\left[\sum_{x \in \pi_q} \big(\epsilon_+ r_x + \epsilon_-(1 - r_x)\big)\delta f(x)\right] > 0$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q}\left[\sum_{x \in \pi_q} (\epsilon_+ - \epsilon_-) r_x \cdot \delta f(x) + \epsilon_- \delta f(x)\right] > 0$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q}\left[\sum_{x \in \pi_q} (\epsilon_+ - \epsilon_-) r_x \cdot \delta f(x)\right] > 0$$

$$\Leftrightarrow \mathbb{E}_{\mathbf{r}_q}\left[\sum_{x \in \pi_q} \delta f(x) \cdot r_x\right] > 0 \;\Leftrightarrow\; \boxed{\mathbb{E}[l(S_1, q)] > \mathbb{E}[l(S_2, q)]}$$

Joachims et al. WSDM 2017

Summary of IPW Framework

- Eliminate selection bias
  - Weight each instance with their inverse examination propensity.
  - Loss with IPW can converge to the loss built on true relevance signals.
- Robust to click noise
  - Convergence is invariant with noisy clicks given enough data.

How to get the examination propensity?

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - Dual Learning Algorithm

Propensity Estimation

$=$

and

$$P(c_i = 1) \quad = \quad P(o_i = 1) \quad \bullet \quad P(r_i = 1)$$

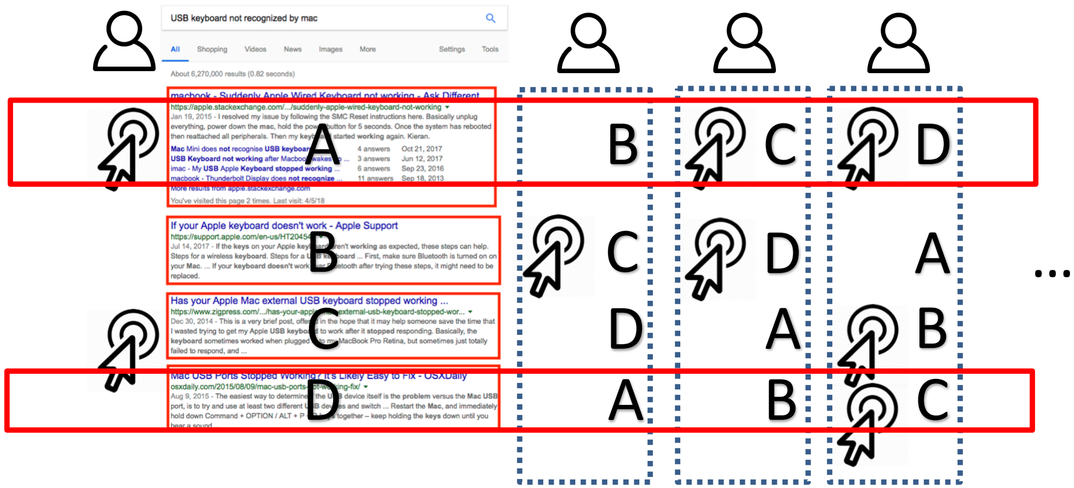
Examination Propensity

Document Relevance

How to remove it?

Online Randomization

- Let $P(r_i = 1)$ become a constant.



$$\forall i, P(r_i = 1) \equiv C$$

Wang et al. SIGIR 2016

Online Randomization

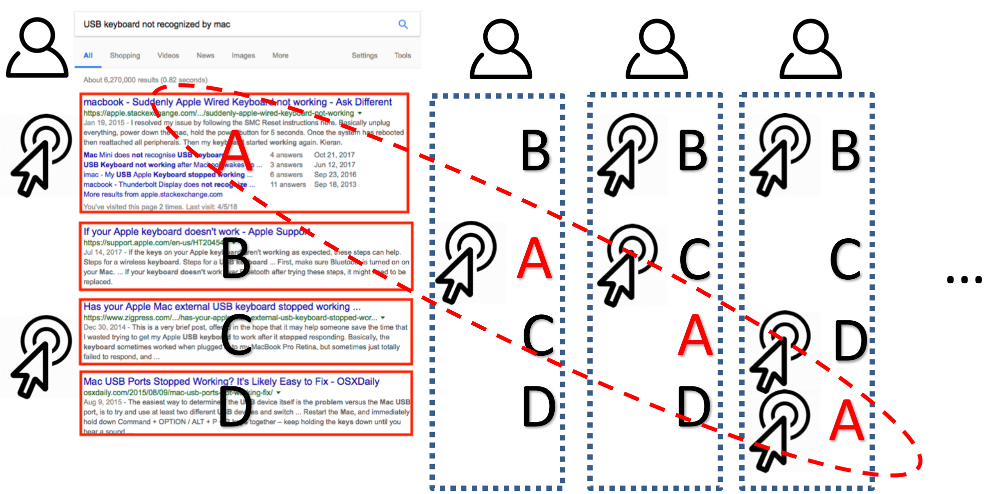
- When $\forall i, P(r_i = 1) \equiv \mathcal{C}$

$$
\begin{aligned}
\mathbb{E}[c_k] &= \int_{(q,x,\pi_q),i=k} P(c_i = 1|\pi_q)\ dP(q,x,\pi_q) \\
&= \int_{(q,x,\pi_q),i=k} P(o_i = 1) \cdot P(r_i = 1|\pi_q)\ dP(q,x,\pi_q) \\
&= P(o_k = 1) \cdot \int_{(q,x,\pi_q),i=k} P(r_i = 1|\pi_q)\ dP(q,x,\pi_q) \\
&\propto P(o_k = 1)
\end{aligned}
$$

Wang et al. WSDM 2018

Online Randomization

- IPW only needs "relative" propensity.
  - Partial result randomization though swapping.

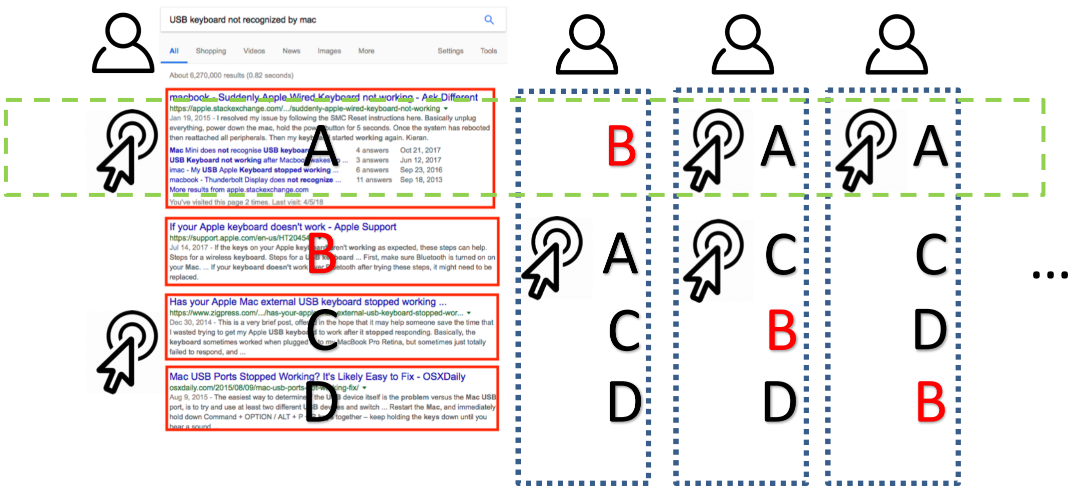


$$\frac{P(o_i=1)}{P(o_1=1)} = \frac{P(c_i=1)/P(r_A=1)}{P(c_1=1)/P(r_A=1)} = \frac{P(c_i=1)}{P(c_1=1)}$$

Joachims et al. WSDM 2017

Online Randomization

- Anchor could be at any position.



Small effect on top-1 results

$$\frac{P(o_i = 1)}{P(o_2 = 1)} = \frac{P(c_i = 1)/P(r_B = 1)}{P(c_2 = 1)/P(r_B = 1)} = \frac{P(c_i = 1)}{P(c_2 = 1)}$$

Joachims et al. WSDM 2017

Summary

- Online result randomization:
  - Effectively estimate the position propensity.
  - Minor effect on search quality if applied properly.

Wang et al. SIGIR 2016

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - Dual Learning Algorithm

Experiments

- Simulation Experiments
  - Training
    - Simulated click data.
  - Testing
    - Human annotated relevance labels.
- Real-world applications
  - Training
    - Clicks from search engine log data.
  - Testing
    - Online A/B test.

Simulation Experiment

- Dataset
  - Yahoo! LETOR (29,921 queries, 710k docs)
  - Five-level relevance annotation (0-4) for each query-document pair.
  - 700 features from production.

	Train	Validation	Test
#Query	19,944	2,994	6,983

Joachims et al. WSDM 2017

Simulation Experiment

- Simulate clicks based on presentation bias:
  - Create initial lists with *SVMrank* (1% training data).
  - Generate clicks on the top 10 results.

Bias severity

$$(\frac{1}{i})^{\boxed{\eta}}$$

Click noise

Annotated relevance label

$$\epsilon_- + (1 - \epsilon_-) \cdot 1_{\boxed{y}>2}$$

$$P(c_i = 1|\pi_q) = P(o_i = 1) \cdot P(r_i = 1|\pi_q)$$

Joachims et al. WSDM 2017

Simulation Experiment

- Propensity estimation
  - The true propensity used to generate synthetic clicks.
- Evaluation metric
  - Averaged rank of relevant documents

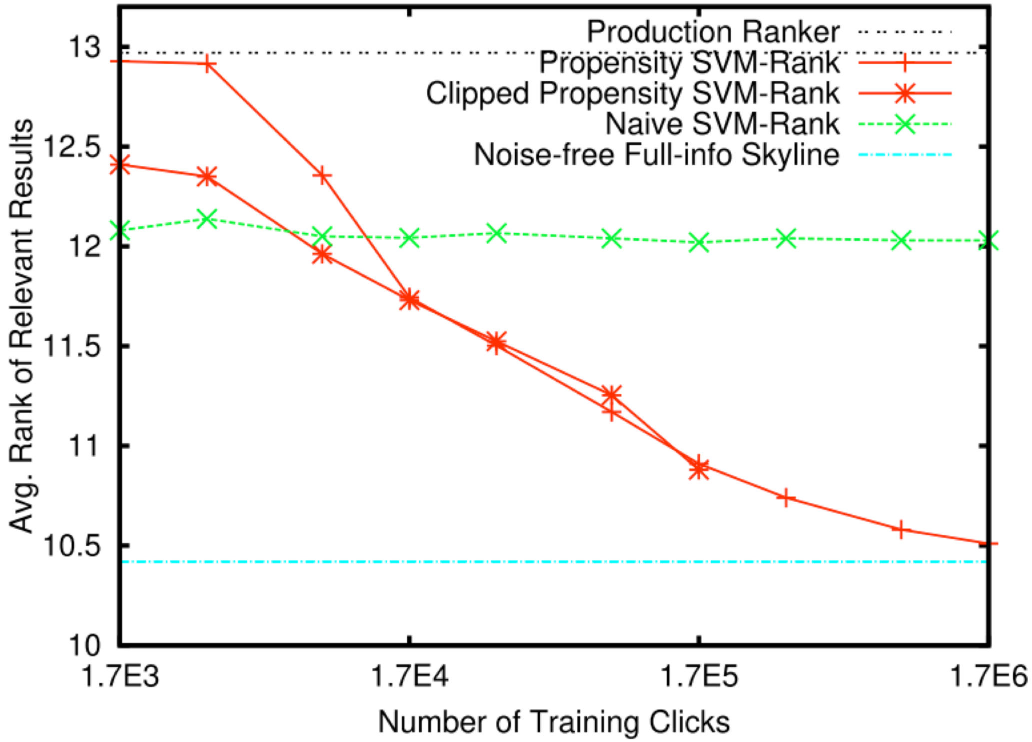
Joachims et al. WSDM 2017

Simulation Experiment

- Baseline
  - *Production Ranker*
    - SVMrank trained with human annotations and 1% training data.
  - *Naïve SVMrank*
    - SVMrank trained with synthetic clicks and all training data.
  - *Noise-free Full-info Skyline*
    - SVMrank trained with human annotations and all training data.
- Treatment
  - *Propensity SVMrank*
    - SVMrank trained with synthetic clicks and IPW.
  - *Clipped Propensity SVMrank*
    - SVMrank trained with synthetic clicks and IPW (clipped weight).

Joachims et al. WSDM 2017

Simulation Results



$\eta = 1$

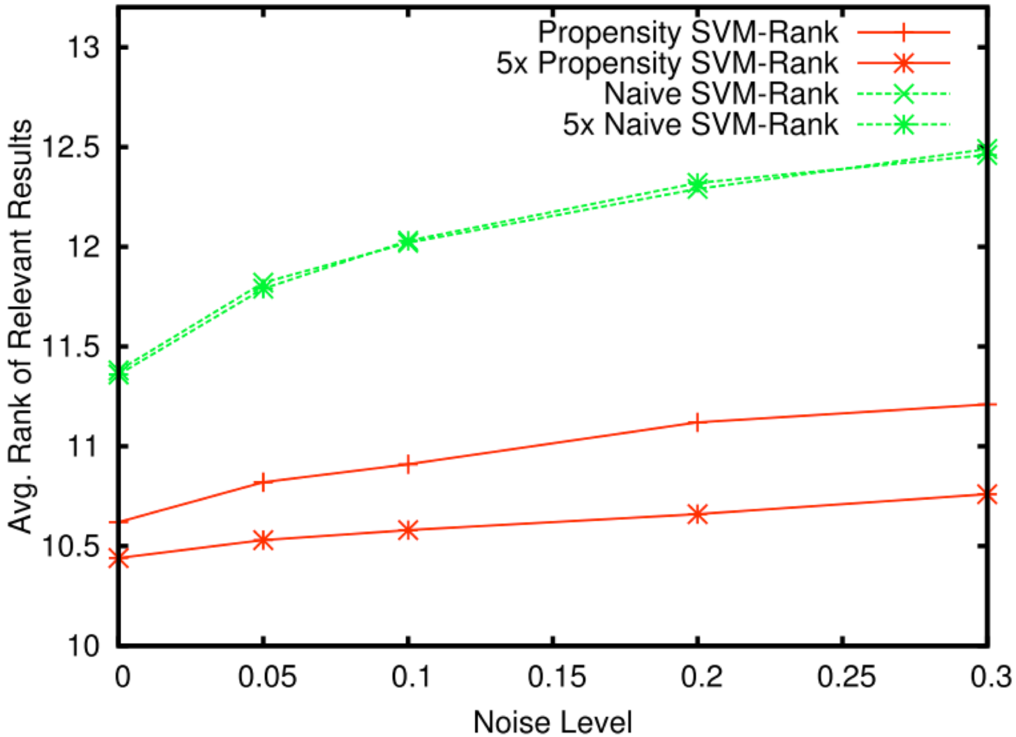
$\epsilon_- = 0.1$

Figure 1: Test set performance of different ranking models with respect to the number of training clicks. The y-axis is the averaged rank of relevant results, which is the lower the better.

- *Propensity SVMrank* outperforms *naïve SVMrank*.
- *Propensity SVMrank* approximates the optimal unbiased ranker given enough training clicks.

Joachims et al. WSDM 2017

Simulation Results



ϵ_-

$\eta = 1$

Figure 2: Test set performance of different ranking models with respect to the probability of noisy clicks. The y-axis is the averaged rank of relevant results, which is the lower the better.

- *Propensity SVMrank* shows consistent improvements over *naïve SVMrank*.
- Quintuple the size of training data can improve the performance of the *Propensity SVMrank*, but not the *Naïve SVMrank*.

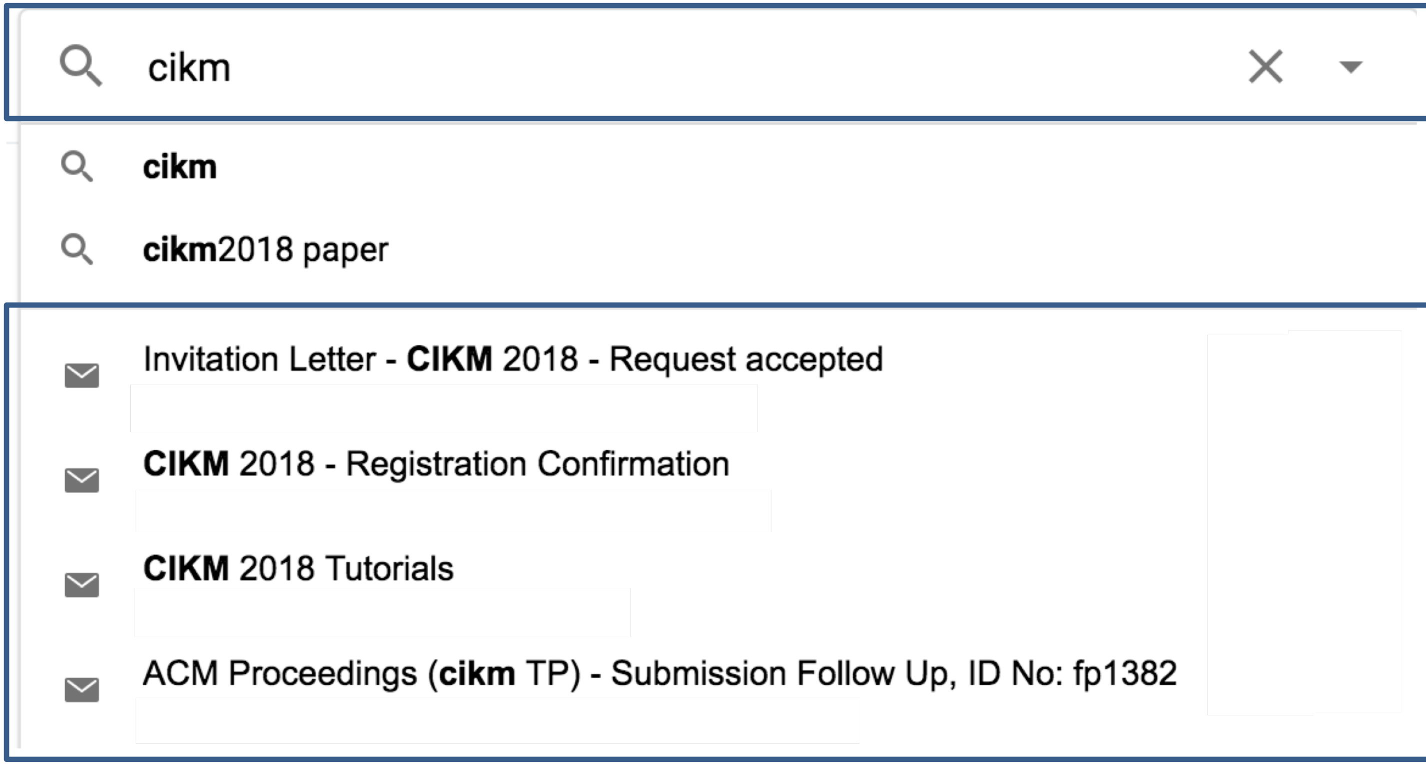
Joachims et al. WSDM 2017

Real-world Applications

- Gmail search

Query

Results



Wang et al. SIGIR 2016

Real-world Applications

- Dataset
  - Regular data
    - Gmail search logs collected from 2015-12-01 to 2015-12-07.
    - 4+ million queries with clicks
  - Randomized data
    - A small fraction of Gmail search traffic from 2015-11-18 to 2015-11-23.
    - 148K queries with clicks.
    - Each query has exactly 4 results shown to the users.

Wang et al. SIGIR 2016

Real-world Applications

- Propensity estimation
  - Results were shuffled in each query of the randomized data before shown to the users.
  - Compute position propensity based on user clicks on the shuffled results.
- Evaluation
  - Mean Reciprocal Rank (MRR), CTR

$$MRR = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{rank_q}$$

$rank_q$: the position of the first clicked document in q.

Wang et al. SIGIR 2016

Real-world Applications

- Ranking model
  - An internal learning-to-rank model similar to GBDT.
- Baseline
  - *NoCorrection*
    - The LTR model trained directly with user clicks.
- Treatment
  - *Global*
    - Compute IPW with all queries in the randomized data.
  - *Segmented*
    - Partition queries into multiple segments (*Social*, *Promotional*, etc.), and compute IPW in each group separately.
  - *Generalized*
    - Train a regression model on the randomized data to predict the IPW of each position based on query features.

Wang et al. SIGIR 2016

Online A/B Test Results

Table 1: The MRR improvement (percentage) of each model with respect to the baseline model. Notation *, **, *** means significant difference at p<0, p<0.05, p< 0.01, respectively

Baseline	Global	Segmented	Generalized
NoCorrection	0.67%***	0.88%***	0.79%***
Global	----	0.21%*	0.12%

Table 2: The CTR improvement (percentage) of each model with respect to the baseline model. Notation *, **, *** means significant difference at p<0, p<0.05, p< 0.01, respectively

Baseline	Global	Segmented	Generalized
NoCorrection	0.46%***	0.71%***	0.62%***
Global	----	0.25%*	0.15%

- IPW significantly improves the MRR and CTR of the online system.
- Query-dependent IPW performs even better.

Wang et al. SIGIR 2016

Summary

- Simulation/real-world experiments:
  - IPW framework can effectively learn an unbiased LTR model with biased supervision signals.
  - It is robust to click noise.
  - It improves the quality of an online system.

Wang et al. SIGIR 2016

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - Dual Learning Algorithm

Advanced Topics

- The pipeline to build an unbiased ranker:

  Day 1: Estimate propensity with online randomization.

  Day 2: Collect users clicks from search engine.

  Day 3: Learn an unbiased LTR model with user clicks collected in day 2 and the IPW estimated in day 1.

  Day 4: System online.

What if the propensity changes from Day 1 to Day 2?

Advanced Topics

- Examination propensity is likely to change:
  - Layout changes
  - Interface changes
  - Presentation changes
  - …

Effect of Misspecified Propensity

- Remember in the simulation experiments:

Bias severity

$$\left(\frac{1}{i}\right)^{\boxed{\eta}}$$

Click noise

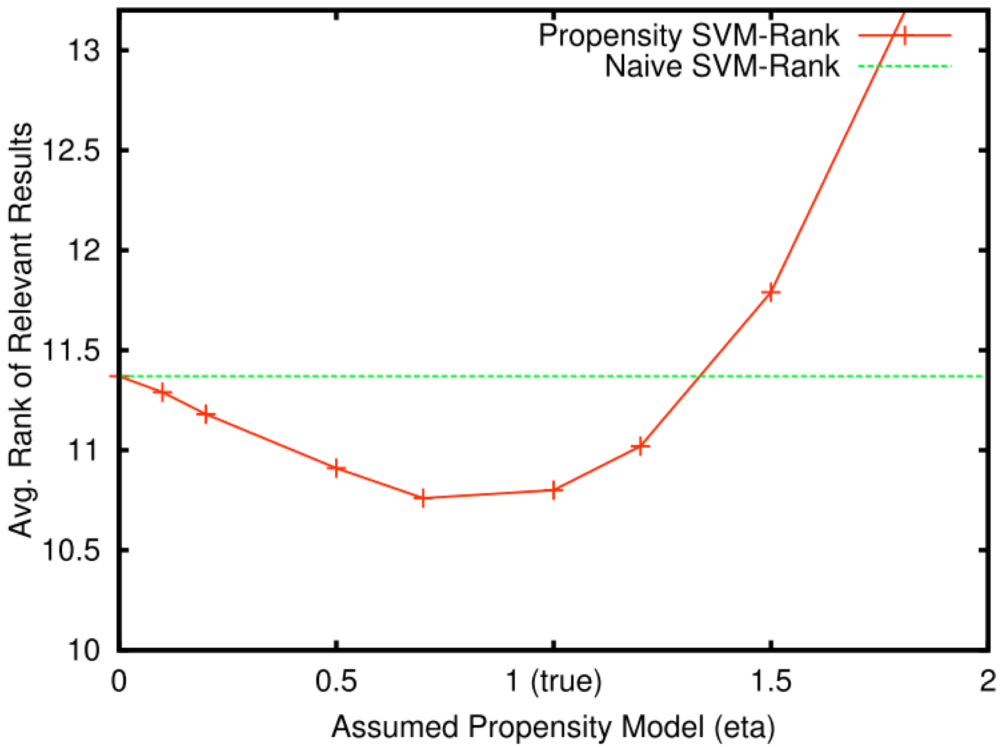
Annotated relevance label

$$\epsilon_{-} + (1 - \epsilon_{-}) \cdot 1_{y>2}$$

$$P(c_i = 1 | \pi_q) = P(o_i = 1) \cdot P(r_i = 1 | \pi_q)$$

Joachims et al. WSDM 2017

Effect of Misspecified Propensity



η

In online randomization:

$\eta = 1$

$\epsilon_- = 0.1$

Figure 3: Test set performance of different ranking models with respect to different bias severity (η). η is 1 when we generate the training clicks.

- *Propensity SVMrank* produces suboptimal performance when the propensity used in IPW is different from those in the training data.

Joachims et al. WSDM 2017

Automatic Unbiased LTR

- Directly learn an unbiased ranker from clicks.
- No online result randomization.
- Adaptive to the changes of examination propensity.

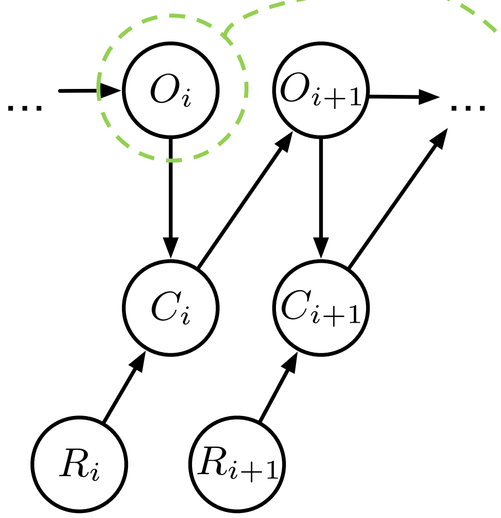
Automatic Unbiased LTR

- The key is to automate propensity estimation.
- Solutions:
  - EM algorithm [Wang et al. WSDM 2018]
  - Dual learning algorithm [Ai et al. SIGIR 2018]

Propensity Estimation with EM

- In click models, we estimate *P(o=1)* from clicks directly.
  - *Why not borrow it for unbiased LTR algorithms?*

Click Model



Inverse Propensity Weighting

$$l_{IPW}(S,q) = \sum_{x_i \in \pi_q, c_i = 1} \frac{\Delta(x_i, c_i|\pi_q)}{\boxed{P(o_i = 1|\pi_q)}}$$

Wang et al. WSDM 2018

Experiments

- Dataset
  - Gmail/Google Drive search logs
    - Collected from two weeks in April 2017.
    - 8+ million queries with clicks
    - Each query has exactly 5 results shown to the users.

Wang et al. WSDM 2018

Experiments

- Propensity estimation
  - Construct a click model with EM algorithm.
  - Extract the propensity parameters.
- Evaluation
  - Weighted Mean Reciprocal Rank (WMRR)

$$WMRR = \frac{1}{\sum_{q \in Q} w_q} \sum_{q \in Q} \frac{w_q}{rank_q}$$

$rank_q$: the position of the first clicked document in q.

w_q: the inverse propensity of the position of the first clicked document in q.

* w_q is estimated by online result randomization.

Wang et al. WSDM 2018

Experiments

- Baseline
  - *NoCorrect*
    - A pointwise GBDT trained directly with user clicks.
- Treatment
  - *Embeded*
    - Add position information as a feature for the pointwise GBDT.
  - *EM*
    - Construct a click model.
    - Estimated the relevance parameters with a pointwise GBDT.
  - *EMCorrected*
    - Construct a click model.
    - Use the propensity parameters from the click model to conduct IPW for a pairwise GBDT.

Wang et al. WSDM 2018

Ranking Performance

Table 3: The WMRR improvement (percentage) of each model with respect to NoCorrect model on the Email dataset. Notation * means significant difference.

	Pointwise		**Pairwise**
Model	Embeded	EM	EMCorrected
WMRR Improvement	-4.44%*	+0.50%+	+1.30%+

- Embedding position features won't work for ULTR directly.
- EM could reduce click bias in some extents.
- Propensity estimation with EM + pairwise ranking model with IPW is a better choice.

Wang et al. WSDM 2018

Propensity Estimation

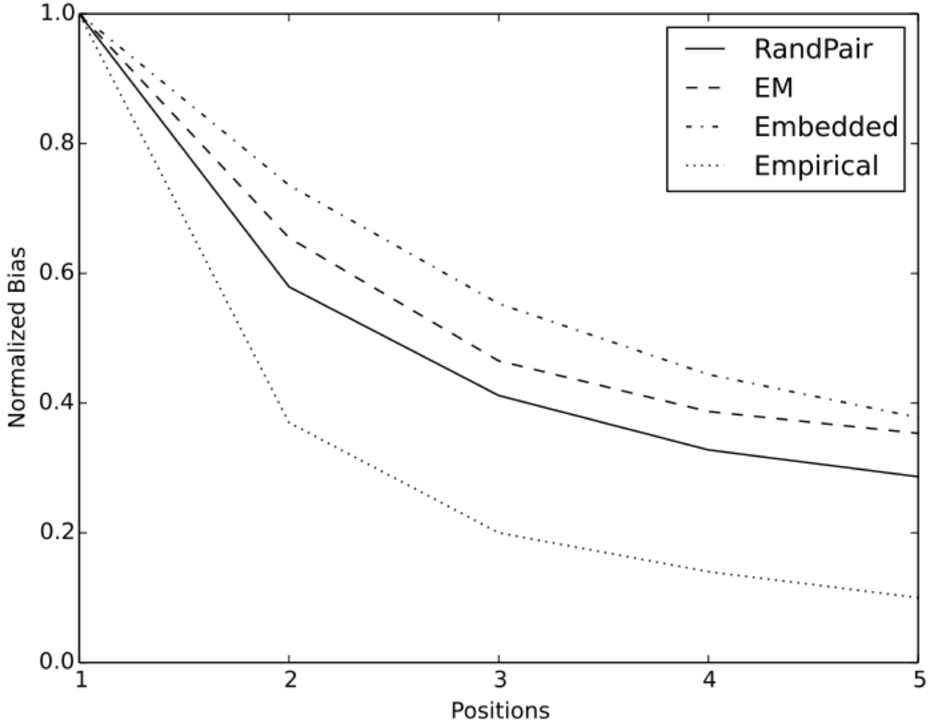


Figure 4: Estimated position bias on the email dataset normalized by the top position.

- *RandPair*: position propensity estimated by online result randomization.
- *Empirical*: position propensity directly computed with CTR.

Wang et al. WSDM 2018

Outline

- Motivation
  - Selection bias
- Inverse Propensity Weighting
  - Proof of correctness
  - Proof of robustness
- Randomized Experiments
  - Online result randomization
  - Pairwise result randomization
- Simulation Experiments
  - Dataset and synthetic clicks
  - Results
- Real-world Experiments
  - Gmail search
  - Results
- Advanced Topics
  - Automatic unbiased learning to rank
    - Propensity estimation with EM
    - **Dual Learning Algorithm**

Re-Visit the Problem

- Step back a little bit:

$=$

and

$$P(c_i = 1) = P(o_i = 1) \cdot P(r_i = 1)$$

$P(c_i = 1)$ and $P(o_i = 1)$ $\Rightarrow$ $P(r_i = 1)$

Inverse Propensity
Weighting

Ai et al. SIGIR 2018

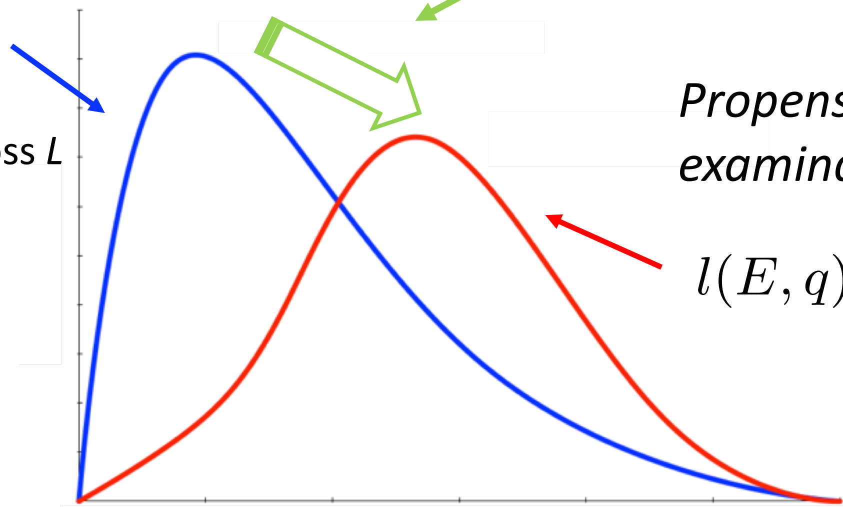
Unbiased Propensity Estimation

- Goal:
  - Estimate the true propensity model E

Propensity loss based on clicks:

$$\hat{l}(E,q) = \sum_{x_i \in \pi_q, c_i = 1} \Delta(x_i, c_i | \pi_q)$$

?
Inverse Relevance Weighting!



Propensity loss based on examination labels:

$$l(E,q) = \sum_{x_i \in \pi_q, o_i = 1} \Delta(x_i, o_i | \pi_q)$$

Unbiased Propensity Estimation

- Inverse Relevance Weighting:

Propensity loss based on clicks

$$l_{IRW}(E,q) = \sum_{x_i \in \pi_q, c_i = 1} \frac{\Delta(x_i, c_i|\pi_q)}{\boxed{P(r_i = 1|\pi_q)}}$$

Propensity loss based on observation labels

$$l(E,q) = \sum_{x_i \in \pi_q, o_i = 1} \Delta(x_i, o_i|\pi_q)$$

$$\begin{aligned}
\mathbb{E}_{\mathbf{r}_q}[l_{IRW}(E,q)] &= \mathbb{E}_{\mathbf{r}_q}\Big[\sum_{x_i \in \pi_q, o_i=1, r_i=1} \frac{\Delta(x_i, o_i|\pi_q)}{P(r_i = 1|\pi_q)}\Big] \\
&= \mathbb{E}_{\mathbf{r}_q}\Big[\sum_{x_i \in \pi_q, o_i=1} \frac{r_i \cdot \Delta(x_i, o_i|\pi_q)}{P(r_i = 1|\pi_q)}\Big] \\
&= \sum_{x_i \in \pi_q, o_i=1} \mathbb{E}_{\mathbf{r}_q}[r_i] \cdot \frac{\Delta(x_i, o_i|\pi_q)}{P(r_i = 1|\pi_q)} \\
&= \sum_{x_i \in \pi_q, o_i=1} P(r_i = 1|\pi_q) \cdot \frac{\Delta(x_i, o_i|\pi_q)}{P(r_i = 1|\pi_q)} \\
&\boxed{= \sum_{x_i \in \pi_q, o_i=1} \Delta(x_i, o_i|\pi_q)}
\end{aligned}$$

$c_i{=}1 \Rightarrow o_i{=}1, r_i{=}1$

How to get the relevance probability?

From ranking model!

Dual Learning Algorithm

- Jointly learn the propensity model and the ranking model:

$$l_{IRW}(E,q) = \sum_{x_i \in \pi_q, c_i = 1} \frac{\Delta(x_i, c_i|\pi_q)}{\boxed{P(r_i = 1|\pi_q)}} \Rightarrow E \Rightarrow P(o_i = 1|\pi_q)$$

Propensity Model

$$l_{IPW}(S,q) = \sum_{x_i \in \pi_q, c_i = 1} \frac{\Delta(x_i, c_i|\pi_q)}{\boxed{P(o_i = 1|\pi_q)}} \Rightarrow S \Rightarrow P(r_i = 1|\pi_q)$$

Ranking Model

Ai et al. SIGIR 2018

Simulation Experiments

- Dataset
  - Yahoo! LETOR (29,921 queries, 710k docs)
  - Train models with clicks, but test models with human annotations.
  - Simulate clicks based on presentation bias:

Presentation bias
[Joachims et al. SIGIR 2005]

Bias severity

$$\rho_i^{\boxed{\eta}}$$

Click noise

Annotated relevance label

$$\epsilon + (1-\epsilon)\frac{2^{\boxed{y}}-1}{2^{y_{max}}-1}$$

$$P(c_i = 1|\pi_q) = P(o_i = 1) \cdot P(r_i = 1|\pi_q)$$

Ai et al. SIGIR 2018

Simulation Experiments

- Baselines
  - *NoCorrect*:
    - Train LTR models with clicks directly.
  - *RandList*:
    - Estimate examination propensity with randomization experiments.
    - Train LTR models with clicks and inverse propensity weighting.
  - *Initial Ranker*
    - The model that generates the initial ranked lists for click simulation.
  - *Oracle Ranker*
    - The model trained with human annotations.
- Treatment
  - *DLA*:
    - Train LTR models with DLA and clicks directly.

Simulation Experiments

Table 4: The performance of different bias correction method in the simulation experiments. Notation +/- means significant better or worse comparing to DLA.

Correction Method	NDCG@1	ERR@1	NDCG@5	ERR@5	NDCG@10	ERR@10
DLA	**0.658**	**0.338**	**0.683**	**0.433**	**0.729**	**0.447**
NoCorrect	0.622 -	0.317 -	0.653 -	0.416 -	0.704 -	0.431 -
RandList	**0.658**	**0.338**	0.679 -	**0.433**	0.725 -	**0.447**
Initial Ranker	0.559 -	0.271 -	0.617 -	0.381 -	0.675 -	0.397 -
Oracle Ranker	0.667 +	0.339 +	0.695 +	0.435 +	0.740 +	0.449 +

- Static mode:
  - Click bias doesn't change over time.
  - In this case, *RandList* is an optimal propensity estimator in theory.

Simulation Experiments

Figure 5: Ranking performance

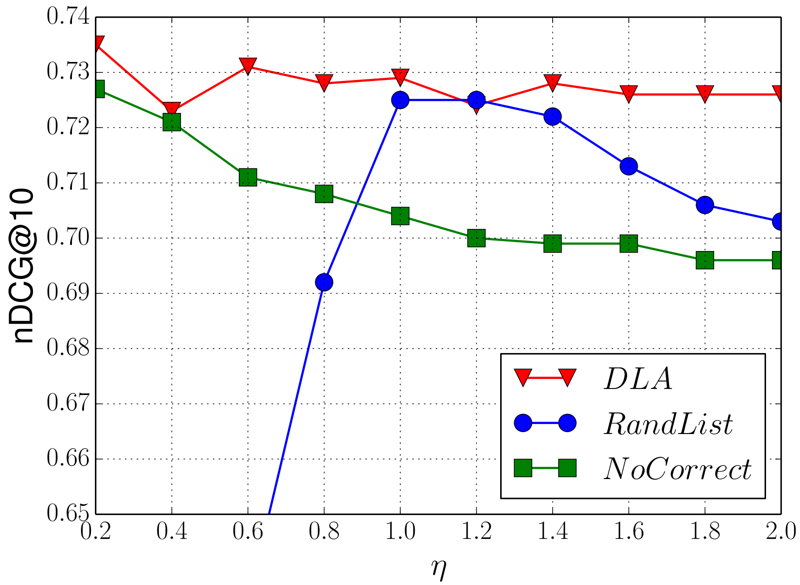
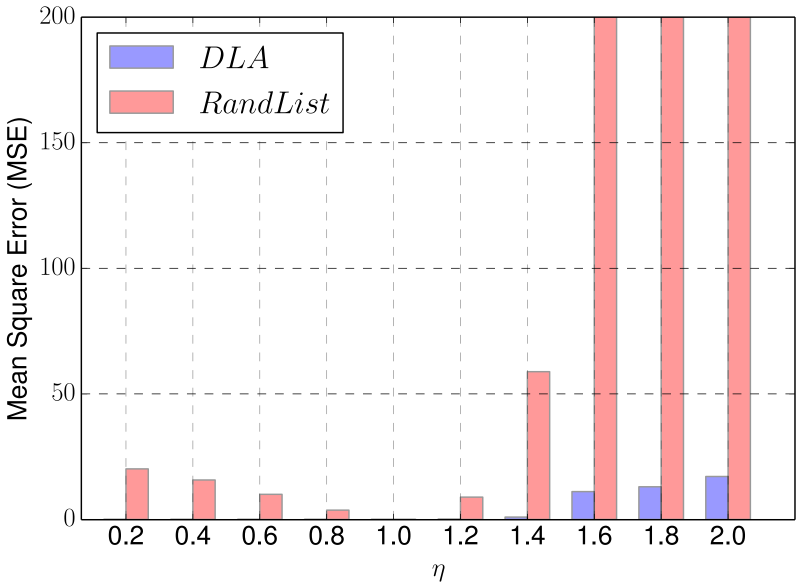


Figure 6: Propensity estimation performance



- Dynamic mode:
  - Click bias changes over time.
  - The presentation bias in the randomization experiment (η=1) could be different from those in model training ($\eta \neq 1$).

Simulation Experiments

Figure 5: Ranking performance

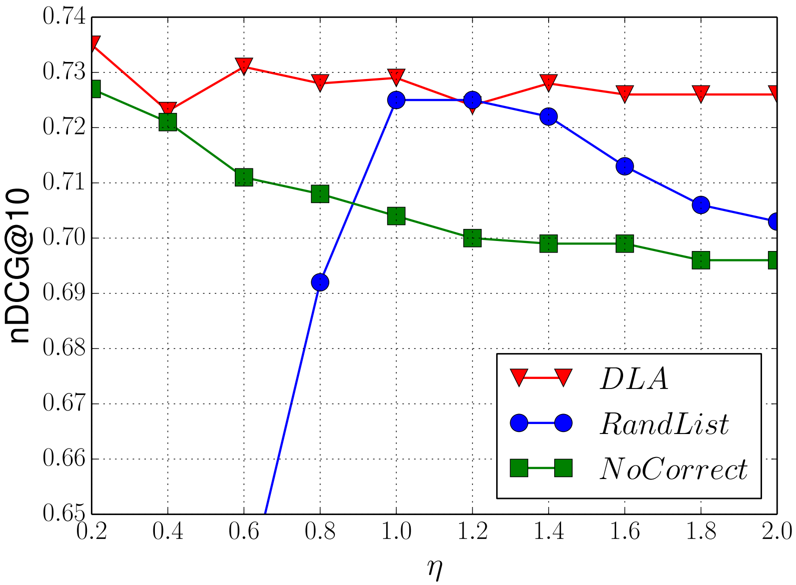
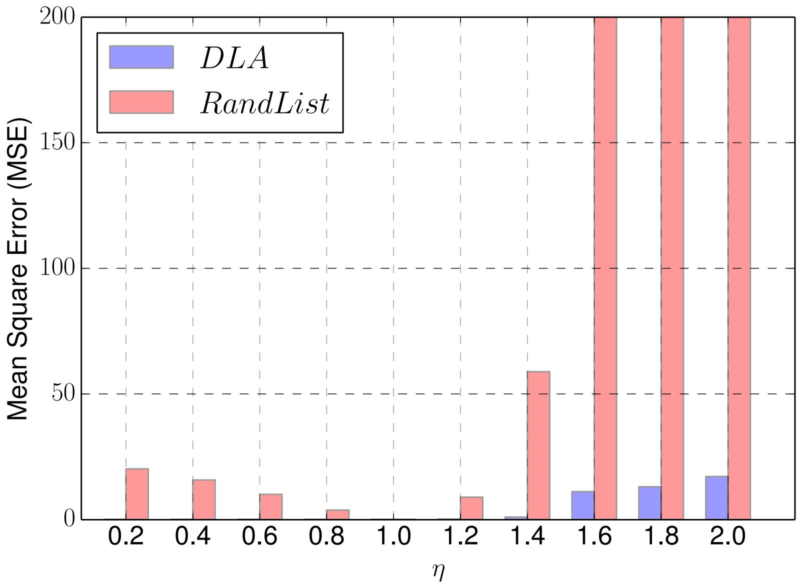


Figure 6: Propensity estimation performance



- Observation:
  - The change of click bias severely hurts the performance of RandList.
  - DLA is much more robust and adaptive.

No result randomization needed !

Real-world Experiments

- Dataset
  - Search logs from a commercial search engine (3,449 queries).
  - Real user clicks (> 3M).
  - LETOR style features [Qin et al. 2010].
  - A separate test collection with 100 queries and 10k docs annotated by human [Zheng et al. 2018].

Real-world Experiments

- Baselines:
  - *NoCorrect*:
    - Train LTR models with click directly.
  - *UBM*:
    - Train LTR models with the relevance signals extracted by UBM [12].
  - *DBN*:
    - Train LTR models with the relevance signals extracted by DBN [13].
- Treatment
  - *DLA*:
    - Train LTR models with DLA and clicks directly.

Real-world Experiments

Table 5: The performance of different bias correction methods in the real-world experiments. Notation +/- means significant better or worse comparing to DLA.

Correction Method	NDCG@1	ERR@1	NDCG@5	ERR@5	NDCG@10	ERR@10
DLA	**0.433**	**0.406**	**0.422**	**0.571**	**0.421**	**0.582**
DBN	0.363	0.340 -	0.390	0.504 -	0.419	0.521 -
UBM	0.359 -	0.336 -	0.352 -	0.502 -	0.365 -	0.519 -
NoCorrect	0.357 -	0.334 -	0.349 -	0.484 -	0.358 -	0.500 -

- Observations:
  - Click models > NoCorrect.
  - **Joint learning > separate learning**.

Outline

- Introduction
  - Problem Analysis
  - Existing Solutions
- Part 1: Click Models
  - Basic Concept and Hypothesis
  - Advanced Click Model
  - Applications
- Part 2: Unbiased Learning Algorithms
  - Inverse Propensity Weighting
  - Online Result Randomization
  - Simulation/ Real-world Experiments
  - Advanced Topics
- Summary

Recap

- Unbiased Learning to Rank
  - Learn an unbiased ranker with biased user feedback.
- Solution 1: Click Models
  - Click bias
  - Behavior Hypothesis
  - Applications
- Solution 2: Unbiased Learning Algorithms
  - Inverse Propensity Weighting
  - Online Result Randomization
  - Simulation and Real-world Applications

Future Study

- Modeling relevance from multiple perspective.
- Unbiased learning with more than position bias.
- …

Related Materials

- Tutorials:
  - *Click Models for Web Search and their Applications to IR* [Chuklin et al. WSDM 2016]
  - *Counterfactual Evaluation and Learning* [Swminathan and Joachims SIGIR 2016]
- Codebase:
  - Click Models
    - https://github.com/THUIR/PSCMModel
  - Propensity SVMrank
    - https://www.cs.cornell.edu/people/tj/svm_light/svm_proprank.html
  - Dual Learning Algorithm
    - https://github.com/QingyaoAi/Unbiased-Learning-to-Rank-with-Unbiased-Propensity-Estimation

Thanks!

Q&A

aiqy@cs.umass.edu, maojiaxin@gmail.com

Reference

- Richardson, Matthew, Ewa Dominowska, and Robert Ragno. "Predicting clicks: estimating the click-through rate for new ads." *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007.
- Joachims, Thorsten, et al. "Accurately Interpreting Clickthrough Data as Implicit Feedback." (2005).
- Craswell, Nick, et al. "An experimental comparison of click position-bias models." *Proceedings of the 2008 international conference on web search and data mining*. ACM, 2008.
- Dupret, Georges E., and Benjamin Piwowarski. "A user browsing model to predict search engine click data from past observations." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.
- Chapelle, Olivier, and Ya Zhang. "A dynamic bayesian network click model for web search ranking." *Proceedings of the 18th international conference on World wide web*. ACM, 2009.
- Wang, Chao, et al. "Incorporating vertical results into search click models." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013.
- Mao, Jiaxin, et al. "Constructing Click Models for Mobile Search." Proceedings of the 41st international ACM SIGIR conference on Research and development in information retrieval. ACM, 2018.
- Liu, Yiqun, et al. "Time-aware click model." *ACM Transactions on Information Systems (TOIS)* 35.3 (2017): 16.
- Liu, Zeyang, et al. "Enhancing click models with mouse movement information." Information Retrieval Journal 20.1 (2017): 53-80.

Reference

- Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In Proceedings of the 40th ACM SIGIR (SIGIR ’17).
- Wang, Xuanhui, et al. "Position Bias Estimation for Unbiased Learning to Rank in Personal Search." (2018).
- Joachims, Thorsten, Adith Swaminathan, and Tobias Schnabel. "Unbiased learning-to-rank with biased feedback." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017.
- Wang, Xuanhui, et al. "Learning to rank with selection bias in personal search." Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. ACM, 2016.
- Joachims, Thorsten, et al. "Accurately interpreting clickthrough data as implicit feedback." ACM SIGIR Forum. Vol. 51. No. 1. Acm, 2017.
- Liu, Tie-Yan, et al. "Letor: Benchmark dataset for research on learning to rank for information retrieval." Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval. Vol. 310. ACM Amsterdam, The Netherlands, 2007.
- Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, and Shaoping Ma. 2018. Sogou-QCL: A New Dataset with Click Relevance Label. In Proceedings of the 41th ACM SIGIR. ACM.
- Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In Proceedings of the 31st ACM SIGIR. ACM, 331–338.
- Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In Proceedings of the 18th WWW. ACM, 1–10.